Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет або центр післядипломної освіти,
(повна назва)

Кафедра програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

Дослідження методів оцінок ієрархічності та різнорідності інтерфейсу веб-сторінок за допомогою нейромережевих технологій
(тема)

Виконав:
студент (ка) 2 курсу, групи ІПЗздм-22-1

Козел О.Д.
(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення
(код і повна назва спеціальності)
Тип програми освітньо-наукова

Керівник доц. Колесніков Д.О.
(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри _____ З.В.Дудар
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет центр післядипломної освіти
Кафедра програмної інженерії
Рівень вищої освіти другий (магістерський)
Спеціальність 121 – Інженерія програмного забезпечення
Тип програми освітньо-наукова програма
Освітня програма Інженерія програмного забезпечення
(шифр і назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«___» _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Козелу Олексію Дмитровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження методів оцінок ієрархічності та різнорідності інтерфейсу веб-сторінок за допомогою нейромережевих технологій
Затверджена наказом по університету від 22.02.2024р. № 60Стз
2. Термін подання студентом роботи до екзаменаційної комісії 14.06.2024
3. Вихідні дані до роботи інформація щодо нейронних мереж
4. Перелік питань, що потрібно опрацювати в роботі
теоретичні основи оцінки юзабіліті інтерфейсу веб-сторін, проектування веб-додатку для оцінки ієрархічності та різнорідності інтерфейсу веб-сторінок за допомогою нейромережних технологій, проектування веб-додатку для оцінки ієрархічності та різнорідності інтерфейсу веб-сторінок за допомогою нейромережних технологій, Реалізація веб-додатку для оцінки ієрархічності та різнорідності інтерфейсу веб-сторінок за допомогою нейромережних технологій

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі	01.04.2024	виконано
2	Виявлення проблематики галузі	08.04.2024	виконано
3	Здійснення огляду математичних моделей	15.04.2024	виконано
4	Розробка алгоритму передобробки даних	22.04.2024	виконано
5	Дослідження можливості прискорення	29.04.2024	виконано
6	Побудова плану експерименту	06.05.2024	виконано
7	Імплементація прототипів обраних моделей	13.05.2024	виконано
8	Аналіз результатів експерименту	20.05.2024	виконано
9	Написання пояснювальної записки	27.05.2024	виконано
10	Підготовка доповіді та презентації	27.05.2024	виконано
11	Підготовка роботи для перевірки на антиплагіат та проходження нормоконтролю	31.05.2024	виконано
12	Оцінка роботи рецензентами	08.06.2024	виконано
13	Отримання відзиву від керівника роботи	10.06.2024	виконано
12	Попередній захист кваліфікаційної роботи	10.06.2024	виконано
13	Здача роботи у електронний архів	10.06.2024	виконано
14	Отримання допуску до захисту	12.06.2024	виконано
15	Захист кваліфікаційної роботи	14.06.2024	виконано

Дата видачі завдання 01 квітня 2024 р.

Студент _____ Козел О.Д.
(підпис)

Керівник роботи _____ доц. Колесніков Д.О.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до кваліфікаційної роботи містить 103 сторінки, 26 рисунків, 3 таблиці, 30 джерел, 6 додатків.

ГЕНЕРАТИВНІ МОДЕЛІ, ГЕНЕРАЦІЯ, НЕЙРОННА МЕРЕЖА, МАШИННЕ НАВЧАННЯ, ПАМ’ЯТЬ, ТЕКСТ.

Основними цілями дослідження є з'ясування проблем юзабіліті цільової інтерактивної системи для того, щоб сформулювати необхідні пропозиції щодо покращення веб-сайту та запропонувати рішення шляхом редизайну існуючої інтерактивної системи.

Об'єктом дослідження є генеративні системи з асоціативною пам'яттю.

Метою роботи є розробка системи оцінки ієрархічності та гетерогенності інтерфейсу веб-сторінок з використанням нейромережевих технологій.

GENERATIVE MODELS, GENERATION, NEURAL NETWORK, MACHINE LEARNING, MEMORY, TEXT.

The main objectives of the study are to find out the usability problems of the targeted interactive system in order to list out required suggestions to improve the website and to provide solutions by re-designing the existing interactive system. In order to implement the objectives of the project, we should initially evaluate the interactive system using usability evaluation.

Objects of research are generative systems with associative memory.

Purpose is a development of a system for evaluating the hierarchy and heterogeneity of the interface of web pages using neural network technologies.

Умови публікації пояснювальної записки

Я, Козел Олексій Дмитрович
(прізвище, ім’я, по батькові)
студент групи ІПЗздм-22-1 здобувач вищої освіти на другому (магістерському) рівні
кафедра програмної інженерії,
(повна назва кафедри)
заявляю: моя кваліфікаційна робота на тему
Дослідження методів оцінок ієрархічності та різнорідності інтерфейсу веб-сторінок за допомогою нейромережевих технологій
(назва роботи)
що буде представлена до ЕК для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату і вона може бути опублікована в електронному архіві відкритого доступу ElArKhNURE. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений (а) з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

Вступ 8
1 Теоретичні основи оцінки юзабіліті інтерфейсу веб-сторін 11
1.1 Опис предметної області 11
1.2 Сучасні методи оцінки юзабіліті веб-сайтів 11
1.3 Аналіз методів оцінки якості інтерфейсів 13
1.4 Аналіз методів оцінки різнорідності елементів 15
1.5 Аналітичний огляд та порівняльний аналіз аналогів 19
1.6 Вибір засобів реалізації 24
2 Проектування веб-додатку для оцінки ієрархічності та різнорідності інтерфейсу веб-сторінок за допомогою нейромережних технологій 27
2.1 Аналіз вимог 27
2.2 Розробка методу оцінки ієрархічності веб-сторінок 29
2.3 Розробка методу оцінки різнорідності веб-сторінок 34
2.4 Алгоритм роботи та поведінкові аспекти системи 35
2.5 Архітектура системи 36
3 Реалізація веб-додатку для оцінки ієрархічності та різнорідності інтерфейсу веб-сторінок за допомогою нейромережних технологій 44
3.1 Підготовка вихідних даних 44
3.2 Підготовка зображення 48
3.3 Розрахунок метрик 50
3.4 Реалізація та навчання нейронних мереж 52
3.5 Реалізація веб-програми 61
3.6 Реалізація серверної частини системи 62
4 Тестування веб-додатку 65
4.1 Функціональне тестування 65
4.2 Інтеграційне тестування 67
4.3 Оцінка коректності роботи системи 67
Висновки 70

Перелік джерел посилання 72
Додаток А. Звіт результатів Перевірки на унікальність тексту в базі ХНУРЕ..... 76
Додаток Б. Реалізація веб-додатку за допомогою нейромережних технологій..... 77
Додаток В. Результати тестування 80
Додаток Г. Презентаційні слайди для захисту кваліфікаційної роботи 83
Додаток Д. Текст наукової публікації за темою кваліфікаційної роботи 96
Додаток Е. Експертний висновок результатів перевірки кваліфікаційної роботи на відповідність оформлення (Вимоги ДСТУ 3008:2015) 103

ВСТУП

Інтернет еволюціонував у середу для широкого спектру дій, включаючи розваги, спілкування, торгівлю, управління, обмін інформацією та інше. Веб-сайт став невід'ємним атрибутом бізнесу: від рітейлерів до виробничих організацій. Соціальні мережі, персональні сайти-візитки, веб-застосунки – яскраві приклади, що використовують веб-сторінки для відображення контенту. За останні п'ять років кількість користувачів Інтернету, а також кількість веб-сайтів значно збільшилась і, як очікується, зростання збережеться тривалий час [1].

З розвитком інформаційних технологій інтерфейс починає грати все більш важливу роль в процесі роботи користувача з інформаційними системами. І на сьогоднішній день, ймовірно, важко назвати такий інформаційний продукт, взаємодія людини з яким здійснюється виключно на рівні інтуїтивного сприйняття, і який не має інтерфейсу користувача. Таким чином, інтерфейс користувача стає основним, а в деяких випадках єдиним каналом для здійснення комунікації між користувачем і веб-сайтом.

Візуально привабливий інтерфейс веб-сайту сприяє створенню позитивного враження від процесу взаємодії користувача з сайтом і, як наслідок, формує у користувача позитивне ставлення до компанії-власника та зміцнює лояльність.

Також привабливий інтерфейс безпосередньо впливає на тривалість використання веб-сайту, служить однією з причин зростання затребуваності веб-сайту і є гарантом його довгострокового використання.

На даний момент оцінка інтерфейсу веб-сайтів проводиться в недостатньому обсязі і її результати не завжди можуть виявити всі проблеми та недоліки, які в кінцевому підсумку можуть вплинути на процес взаємодії користувачів з веб-сайтом.

Важко уявити світ інформаційних продуктів без графічного інтерфейсу: людина звикала взаємодіяти через візуальні елементи додатків. Людино-машинні інтерфейси неухильно вдосконалюються, щоб використання інформаційних систем стало зручним, інтуїтивно зрозумілим та приємним. Стандартизація

інтерфейсів полегшує досягнення цієї мети.

Одним з важливих критеріїв успішного просування бізнесу в мережі інтернет з точки зору досвіду користувача стало залучення і утримання клієнта [2]. У роботі, пов'язаної зі створенням прикладної онтології для оцінки якості користувацьких веб-інтерфейсів, SEO було виділено як один із найважливіших напрямків.

Діяльність вводиться поняття різнорідності елементів, описується його практичне застосування [3].

Інтерфейс користувача з точки зору бізнесу впливає на якість послуг, що надаються, створює позитивне ставлення до веб-сервісу, залишає бажання користуватися надалі. До якості веб-інтерфейсів висуваються підвищені вимоги. Користувальницька перевага відіграє важливу роль [4]. Дослідження підтверджують цю теорію [5,6,7].

Технічна естетика та ергономіка застосовна до веб-середовища та затребувана користувачем. Дослідження на тему якості інтерфейсу користувача і його ергономіки проводяться регулярно [6,8-10] і висуваються нові теорії, використовуються нові інструменти для отримання достовірної інформації.

Проектування, оцінка якості інтерфейсу користувача і його підтримка на ринку послуг в основному представлена у вигляді експертної оцінки. Тренд сучасних досліджень у галузі UI/UX якості – застосування накопичених знань, експертизи та технічного досвіду для створення автоматизованих експертних систем. p align="justify"> Роботи, спрямовані на досягнення такої мети нові, і часто мають дослідницький вузькоспрямований характер [4].

Затребувані інструменти контролю UI/UX якості ще представлені на ринку автоматизованими засобами. Іншими словами, потреби не вкриті продукцією.

Метою курсового проєкту є розробка системи оцінок ієрархічності та різнорідності інтерфейсу веб-сторінок за допомогою нейромережевих технологій. Для досягнення поставленої мети необхідно вирішити такі завдання:

- вивчити предметну область та здійснити порівняльний аналіз існуючих методів оцінок ієрархічності та різнорідності інтерфейсу;

- розробити методику оцінок ієрархічності та різнорідності інтерфейсу веб-сторінок;
- спроєктувати архітектуру системи оцінок ієрархічності та різнорідності веб-сторінок за допомогою нейромережевих технологій.

В курсовому проєкті описуються теоретичні обґрунтування до створення методики для оцінки якості, ґрунтуючись на різнорідності елементів та створення цієї методики. Також описується практична реалізація системи оцінки якості, заснована на актуальних дослідженнях в галузі UI/UX якості, з використанням стандартів ISO, методах оцінки різнорідності компонентів додатків [8,11].

Система, що розробляється, дозволяє на основі роботи нейронної мережі визначити ступінь відповідності інтерфейсу користувача встановленим нормативним характеристикам.

1 ТЕОРЕТИЧНІ ОСНОВИ ОЦІНКИ ЮЗАБІЛІТІ ІНТЕРФЕЙСУ ВЕБ-СТОРІН

1.1 Опис предметної області

Сьогодні оцінка функціональної зручності та візуальної привабливості веб-сайту є дещо суб'єктивною і значною мірою залежить від людського сприйняття.

Через відмінності в особистих уподобаннях і культурних особливостях різні групи користувачів веб-сайтів можуть зробити дуже різні висновки про якість користувацького інтерфейсу. Тому важко провести точну і безпомилкову оцінку юзабіліті за допомогою автоматизованих інструментів.

Реалізація інтерфейсу для роботи з інформаційною системою впливає на успіх цієї системи: користувач зацікавлений у дослідженні функціоналу, отримує естетичне задоволення, відчуває комфорт, якщо реалізація створена на основі загальнокультурних принципів та очікувань. Це впливає як на тривалість взаємодії користувача із системою, так і на рівень задоволеності користувача після взаємодії із системою, і як наслідок, на бажання використовувати систему надалі [6].

Оскільки відсутні інші заходи, які забезпечують високий рівень достовірності, задоволеність користувачів визнано найкориснішим показником успішності системи [10]. Задоволені користувачі проводять більше часу на веб-сайті і часто відвідують його. Загалом, задоволеність користувачів може призвести до утримання аудиторії, підвищення довіри до продукту. Отже, важливо покращувати показники, що підвищують задоволеність веб-сайтом [3].

1.2 Сучасні методи оцінки юзабіліті веб-сайтів

Сучасні методи оцінки юзабіліті є досить широкий перелік методів та засобів, починаючи від інтерв'ювання користувачів та проведення опитувань, закінчуючи використанням складних технічних пристроїв для моніторингу руху очей та систем автоматичної оцінки юзабіліті.

Сучасні методи оцінки юзабіліті можна розділити на наступні категорії:

- методи, засновані на спостереженні за поведінкою користувачів;
- методи, засновані на самооцінці поведінки користувача;
- методи, засновані на непрямому залученні користувачів (див. таблиці 1.1 та 1.2) [1,6].

Таблиця 1.1 – Методи оцінки юзабіліті за участю користувачів

Метод	**Переваги**	**Недоліки**
Лабораторне юзабіліті-тестування	Найнадійніша інформація про взаємодію користувача з інтерфейсом	1) висока чи середня вартість; 2) високі тимчасові витрати; 3) необхідна наявність професійного досвіду
Немодероване юзабіліті-тестування	Найменша вартість порівняно з лабораторним юзабіліті-тестуванням	Зменшення точності та надійності
Айтрекінг	1) точні дані про взаємодію з інтерфейсом; 2) глибокий аналіз дизайну	1) найвища вартість; 2) не замінює інші методи оцінки
Веб-аналітика	1) швидка оцінка; 2) невисока вартість; 3) велике охоплення аудиторії	1) середня точність та надійність; 2) не виявляє причин проблем
Інтерв'ю	1) найдокладніші якісні дані; 2) гнучкість підходу	1) проблеми тлумачення та аналізу; 2) великі тимчасові витрати
Анкетування	Формалізований підхід до виміру задоволеності	Ймовірність помилок у питаннях та відповідях
Опитування	1) швидке отримання якісної оцінки; 2) невисока вартість	1) середня точність та надійність; 2) ймовірність помилок у питаннях та відповідях

Таблиця 1.2 – Методи оцінки юзабіліті з непрямою участю користувачів

Метод	**Переваги**	**Недоліки**
Експертна оцінка	Невисока вартість	1) ймовірність упущення важливих проблем; 2) середні або високі часові витрати
Автоматична оцінка	1) невисока вартість; 2) швидка оцінка	1) ймовірність упущення важливих проблем; 2) експериментальний спосіб
Метод на основі моделей	1) доступність та низька вартість; 2) покращує розуміння процесу взаємодії на етапі прототипування	1) необхідна наявність професійного досвіду для створення персонажів та маршрутів; 2) середні та високі тимчасові витрати

У ситуаціях, коли користувачі беруть безпосередню участь, неправильне тлумачення або неправильні відповіді на запитання та низька надійність можуть

вплинути на достовірність результатів. Експертні методи, автоматизовані оцінки або моделювання процесів не тільки забирають багато часу на реалізацію, але й можуть випускати з уваги важливі питання і проблеми, знижуючи надійність кінцевих результатів оцінки.

Таким чином, на основі аналізу різних інструментів оцінки юзабіліті можна зробити висновок, що жоден з двох існуючих методів оцінки не забезпечує повну, точну і надійну оцінку юзабіліті.

1.3 Аналіз методів оцінки якості інтерфейсів

Методи оцінки юзабіліті поділяються на великі категорії [4]:

а) методи, що передбачають безпосередню участь користувачів:
   1) спостереження за роботою користувачів – збирання інформації про поведінку та дії користувачів у контексті певних завдань під час роботи користувача з програмою;
   2) аналіз критичних подій – збір даних про специфічні події (позитивні або негативні), що відбулися в процесі роботи користувача з програмою;
   3) вимірювання, пов'язані з робочими характеристиками – збір даних про кількісно вимірні робочі характеристики з метою розуміння впливу проблем, пов'язаних з юзабіліті;
   4) анкетування – непрямі методи оцінки, за допомогою яких здійснюється збір думок користувачів про інтерфейс користувача в визначених анкетах;
   5) опитування – здійснюється за аналогією з анкетуванням, але з більшою гнучкістю та при особистому контакті з опитуваною особою;
   6) спільне проектування та оцінка – методи, які уможливлюють співпрацю різних видів учасників при здійсненні оцінки або проектування систем;
   7) метод «Думки вголос» – користувачі постійно вимовляють уголос усі свої думки, переконання, очікування, сумніви, відкриття під час

використання системи, що тестується;

8) творчі методи – методи, які включають виявлення властивостей нових продуктів і систем, які зазвичай отримуються в результаті взаємодій членів групи, при цьому часто членами таких груп є користувачі;

б) методи, які передбачають непряму участь користувачів, які використовуються у разі відсутності можливості зібрати дані про використання через відсутність користувачів, або у випадках, коли вони надають додаткові дані та інформацію:

1) підходи, засновані на використанні моделей – використання моделей, що є абстрактним поданням продукту, що оцінюється, що дозволяють прогнозувати дії користувачів;
2) методи, що ґрунтуються на вивченні документів – вивчення існуючих документів фахівцем з юзабіліті з метою формування професійної оцінки системи;
3) автоматична оцінка – алгоритми, орієнтовані критерії придатності використання чи використання систем, заснованих на знаннях ергономіки, з допомогою яких визначаються недоліки продукту результаті порівняння з визначеними даними;
4) експертна оцінка – оцінка, заснована на знаннях, професіоналізмі та практичному досвіді в галузі ергономіки спеціаліста юзабіліті.

Розглянемо докладніше про методи автоматичної оцінки:

- ентропія RGB-профілю. Оцінюється зорова складність системи;
- інформаційна продуктивність. Відношення мінімальної кількості інформації, необхідної для виконання завдання, до кількості інформації, яку повинен запровадити користувач;
- визначення середнього часу необхідного користувачеві за методикою GOMS, KLM. На основі середніх значень обчислюється обсяг часу в середніх значеннях, які користувач витратив на виконання основних завдань. Користувацькі сценарії визначаються індивідуально для кожного проекту;

– аналіз XML-дерева. Перевіряється складність структури наданої сторінки. Цей метод вимагає спеціалізацію в галузі розробки веб-клієнтів та принципів оптимізації сторінок;
– кількість класів, куди можна розбити об'єкти інтерфейсу.

1.4 Аналіз методів оцінки різнорідності елементів

Різнорідність веб-елементів один із важливих критеріїв, що становлять оцінку задоволеності. Цей критерій впливає простоту засвоєння інформації, сприйняття веб-сторінки, зручність управління системою.

Важливо визначити, що інтерфейси зазвичай служать двом основним цілям [3]:

– надання інформації користувачеві;
– забезпечення взаємодії із системою.

Процес створення інтерфейсів поділяється на два етапи:

– досвід взаємодії (UX) – формує взаємодію;
– інтерфейс користувача (UI) – візуалізує або матеріалізує взаємодію.

Експериментальні дослідження.

У роботі [4] представлені таблиці, в яких описані такі важливі атрибути, як доступна (understability), добре представлена (well-presented) та організована (well-organized) інформація, інтерактивність, навігація (можливість простого переходу між різними сторінками ресурсу), Простота використання (ease of use), які в різній пропорції створюють міру оцінки різнорідності. Їх коротка розшифровка, надана у цій роботі.

Докладніше про методи автоматичної оцінки:

– Understability (доступність) – чіткість та повнота інформації на веб-сайтах;
– Well-presented (добре подана [інформація]) – якісне подання інформації, опублікованої на сайтах;
– Ease of use (Простота використання) – показує, наскільки легко

користувачам використовувати функції веб-сайту;

– Well-organized (добре організована [інформація]) – контрольована (мається на увазі інтуїтивно зрозуміла організація) та структуроване веб-середовище.

Ці критерії використовувалися під час створення версії анкети з метою оцінки різнорідності інтерфейсу сторінок навчання нейронної мережі, заснованої на анкетах користувачів. Виходячи з цих анкет, була побудована нейронна мережа. Результати та вага кожного з параметрів у нейронній мережі визначені на рисунках 1.1, 1.2.

	System quality	Information quality	Security-privacy
Understandability	0.133	**0.854**	0.192
Reliability	0.266	**0.692**	0.253
Usefulness	0.216	**0.853**	0.137
Access	**0.807**	0.150	0.247
Friendliness	**0.828**	0.158	0.174
Navigation	**0.770**	0.282	0.065
Interactivity	**0.653**	0.201	0.480
Privacy	0.293	0.255	**0.795**
Security	0.156	0.206	**0.882**

Note: Bold values indicate the highest influence weight.

Рисунок 1.1 – Ваги нейронної мережі для загальних критеріїв сайту

	System quality	Information quality	Security-privacy
Easy to comprehend	0.099	**0.809**	0.165
Well-presented	0.117	**0.773**	0.172
Accurate	0.263	**0.738**	0.169
Up-to-date	0.271	**0.667**	0.203
Relevant	0.141	**0.817**	0.134
Detailed	0.138	**0.796**	0.192
Speed of access	**0.729**	0.175	0.169
Availability	**0.778**	0.088	0.096
Ease of use	**0.801**	0.174	0.147
Well-organised	**0.794**	0.168	0.192
Page-loading	**0.702**	0.168	0.140
Hyperlinks	**0.749**	0.178	0.091
Two-way communication	**0.655**	0.192	0.425
Active control	**0.606**	0.109	0.456
Confidentiality	0.316	0.267	**0.701**
Authorisation	0.174	0.222	**0.800**
Integrity	0.225	0.218	**0.784**
Protection	0.130	0.185	**0.830**

Note: Bold values indicate the highest influence weight.

Рисунок 1.2 – Ваги нейронної мережі для приватних критеріїв сайту

Як можна побачити, підстійкість, оцінена нейронною мережею вище всіх в розділі якості інформації.

Так само well-presented, well-organized, ease of use високо оцінені навченої нейронної мережею.

В іншій роботі [3], були отримані аналогічні метрики, спрямовані для створення систем оцінки якості інтерфейсу користувача. На рисунку 1.3 представлено схему онтології.

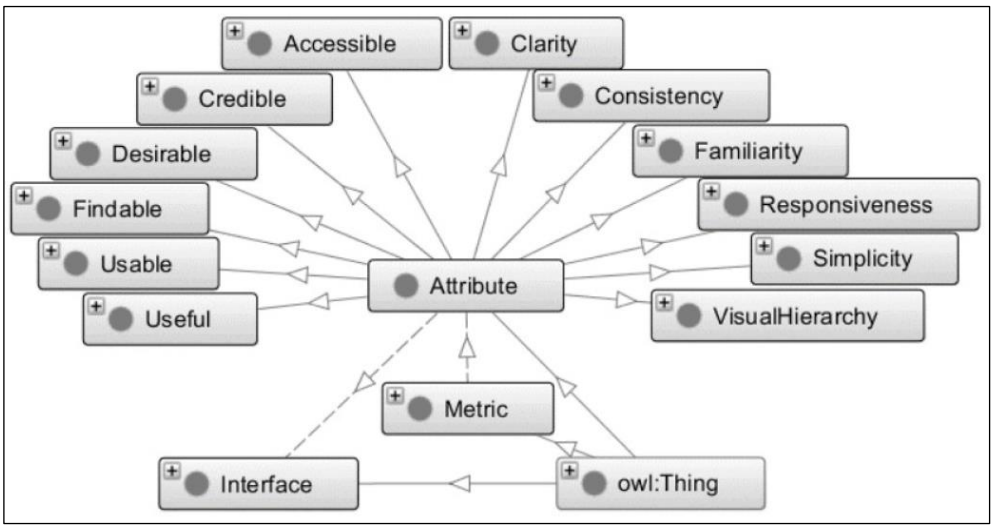


Рисунок 1.3 – Діаграма онтології інтерфейсу користувача

Виходячи із завдання, були визначені напрямки онтології на створення інструменту для оцінки таких атрибутів як: читаність, зручність, чистота, простота [3].

Стандарти та специфікації.

Формалізувати ці аспекти вдалося, використовуючи стандарти ISO, пов'язані з взаємодією людини з дисплеями та інтерфейсами. Вона регулюється стандартами в галузі юзабіліті та людино-машинної взаємодії:

– процес проектування інтерактивних систем, орієнтованих користувачів, містить рекомендації щодо організації процесу проектування інтерфейсів та органічного вбудовування цього процесу у загальний процес виробництва ПЗ. У ньому описані методи юзабіліті, необхідні для: визначення контексту використання продукту, виявлення вимог

користувачів та замовників до системи, прототипування та юзабіліті-тестування продукту;

- ергономіка людино-комп'ютерної взаємодії, опис процесу проектування інтерфейсів, орієнтованих користувачів. У ньому детально описано модель зрілості організації з погляду рівня використання у ній UCD-процесу. Надаються рекомендації щодо переходу на вищі рівні зрілості;
- ергономіка програмного забезпечення мультимедійних інтерфейсів користувача. У ньому надаються рекомендації щодо створення елементів керування для мультимедійних продуктів;
- ергономіка взаємодії "людина-система". Настанови щодо доступу до інтерфейсів "людина-машина";
- ергономіка взаємодії людина-система. Методи, що ґрунтуються на зручності застосування, для забезпечення проектування, орієнтованого на людину.

Оскільки ISO надають лише технічні вимоги до реалізації веб-сторінок, візуальна та варіативна частина дизайну залишилася поза полем вищеописаних стандартів. Важливо відзначити, що стандартами визначається палітра кольорів і нормується її контрастність. Це важливий критерій, який разом різних елементів може вплинути на різнорідність. Також нормуються способи введення/виводу, основи поведінки елементів (наприклад, принципи взаємодії з елементом «кнопка»), їх варіативність.

Дослідження експертів.

Nielsen & Norman Group – велика експертна компанія, що надає послуги оцінки та вдосконалення дизайну у сфері UX/UI.

Для формування експертних методів оцінки також було взято їх публікації. Зокрема, стаття, яка надає більше сотні порад для створення якісної веб-сторінки [12]. Для експертної системи було відібрано деякі рекомендації, що стосуються обраних метриків.

Також у роботах співробітників компанії NN Group описуються закономірності, найкращі рішення та явні помилки у проектуванні елементів сайту

[13-21].

На основі отриманих знань можна виробити основи для методу оцінки різнорідності елементів веб-сторінки:

Підсумком отриманих даних від різних джерел, описаних раніше, можна вважати, що для оцінки різнорідності потрібно взяти до уваги такі критерії як: доступність (виявлення елементів), якість інформації, що надається, простота використання, організація компонент, зручність, чистота, простота.

Приклад реалізації експертної системи з використанням нейронної мережі, здатної оцінити якість сайту, дозволяє зрозуміти, що є практичні підтвердження висунутим метрикам якості. Отримані ваги для критеріїв дають гарне уявлення про цінність кожного критерію оцінки. Виходячи з поставленого завдання можна зрозуміти, що є всі теоретичні передумови створення системи на основі нейронних мереж для оцінки різнорідності елементів.

Підсумком дослідницької частини є вироблене поняття різнорідності інтерфейсу. Різнорідність інтерфейсу – кількість об'єктів та його класів.

1.5 Аналітичний огляд та порівняльний аналіз аналогів

У процесі вивчення аналогів були досліджені веб-додатки з оцінки інтерфейсу користувача, що використовують у своїй роботі скрін-шоти в якості вихідних даних [12].

UsabilityHub – веб-додаток, який дозволяє на основі завантаженого скріншота визначити якість юзабіліті. В рамках аналізу можна отримати інформацію, наскільки користувачам легко орієнтуватися на сторінці веб-сайту, виявити елементи, що привертають найбільшу увагу, а також скласти теплову карту кліків. Отримання оцінки забезпечується за допомогою анкетування користувачів.

UserPlus – веб-додаток, який дозволяє на основі завантаженого скріншота визначити якість юзабіліті. Кожен скріншот розмічується користувачем сервісу самостійно, а потім після опитування публікується результат аналізу юзабіліті для кожного розміченого елемента інтерфейсу.

Usabilla – веб-додаток, який на основі завантажених скріншотів та заздалегідь підготовлених питань забезпечує анкетування користувачів та формує аналітику за його результатами.

ConceptFeedback – інтернет-ресурс, де можливо отримати оцінку інтерфейсу користувача від професійних дизайнерів.

Порівняльний аналіз аналогів наведено у таблиці 1.3.

За результатами порівняльного аналізу можна зробити висновки про те, що більшість веб-додатків використовують як метод оцінки інтерфейсу анкетування та опитування користувачів та тестувальників. Також у ряді програм проводиться моніторинг активності користувача на веб-сторінці. Жоден з наявних аналогів не використовує автоматичну систему оцінки інтерфейсу користувача.

Оцінка інтерфейсу користувача у відповідності з міжнародним стандартом проводиться тільки в додатку UserPlus. Однак цей метод оцінки в основному відноситься до окремих елементів, що становлять інтерфейс, ніж стосується загальної оцінки юзабіліті інтерфейсу.

Таблиця 1.3 – Порівняльний аналіз аналогів систем оцінки юзабіліті веб-сайтів

Назва	**Експерти**	**Інструменти оцінювання**	**Автоматична оцінка**	**Оцінка за стандартом ISO**
Usability-Hub	Тестери сервісу, інші користувачі UsabilityHub	Відповіді на питання, теплова карта кліків	–	–
UserPlus	Тестери сервісу, замовник тестування за допомогою опитувальника	Відповіді на питання, теплова карта кліків	–	+
Usabilla	Користувачі, запрошені замовником тестування	Звіти про дії користувачів, теплова карта кліків	–	–
Concept-Feedback	Учасники спільноти веб-дизайнерів	Відгуки, оцінки та поради учасників професійної спільноти	–	–

За результатами огляду сучасних методів оцінки веб-інтерфейсів, а також на підставі виявлених недоліків серед досліджених аналогів веб-додатків для оцінки інтерфейсів, виникає необхідність у розробці власного методу оцінки

ієрархічності інтерфейсу веб-сторінок та веб-додатку, що забезпечує роботу зі скріншотами на підставі розробленого методу.

Особливості вибору аналогів.

Готових публічних продуктів, доступних користувачам, а також реалізацій з підходом, поставленим у цій роботі, на даний момент немає. Виходячи з цього, вирішено дослідити прототипи, результати теоретичних досліджень, суміжних за напрямом рішень.

У дослідній роботі [4] описані теорії, такі як теорія двох факторів, теорія очікування спростування та трифакторна теорія. Ці теорії стверджують, що вплив атрибуту веб-сайту на задоволеність може мати різні ваги для різних характеристик, що означає, що їхня важливість залежить від їхньої ефективності. Цей факт призводить до нелінійних та асиметричних відносин, які важко оцінити за допомогою традиційних методів. Тому успішні результати досягаються з використанням нейронних мереж, які представлені далі.

Наведені нижче аналоги реалізують деяку частину від поставленого завдання, тому опис кожної – це підтверджений варіант реалізації компонента системи, який належить реалізувати з урахуванням специфіки поставленого завдання.

Огляд експертної системи на основі анкетування.

У роботі [3] робиться спроба визначити відношення загальної задоволеності користувачів до атрибутів веб-сайту. Діяльність використовуються експериментальні результати великого опитування, заснованого на анкетах. Вхідними даними також є анкети. Мета опитування – виявити загальну задоволеність користувачів веб-сайту, відповідаючи на запитання, які стосуються певних атрибутів веб-сайту. У ході цього опитування попросили оцінити набір із 370 користувачів інтернету ефективність 18 конкретних та 9 загальних атрибутів, а також заявити про своє загальне задоволення за дев'ятибальною шкалою з прив'язкою від «дуже незадоволений» до «дуже задоволений». Результати були протестовані та підтверджені за допомогою процедур надійності та валідності, що показують, що існує структура відносин, оскільки певні загальні та специфічні

атрибути веб-сайту створюють зв'язок із задоволеністю користувачів. У цій статті ми намагаємося з'ясувати взаємозв'язок загальної задоволеності конкретними атрибутами веб-сайту за допомогою нейронних мереж для апроксимації функцій.

Результатом дослідницької роботи стала експертна нейронна мережа, що відображає результати цінності кожного з критеріїв на задоволеність користувача залежно від якості виконання цього критерію. Отримані результати зображені на рисунках 1.4–1.7.

Дані рисунків дозволяють зрозуміти цінність кожного з критеріїв у процесі зіставлення зі своїми результатами.

Можна побачити, що це критерії разом дають приріст задоволеності у співвідношенні друг до друга. Виходить, що метод оцінки, що планується, повинен вийде продуктивним при будь-якій якості зображення, а результат оцінки інтуїтивно зрозумілим для користувача системи.

На жаль, система, що описується в досліджуваній роботі, вимагає експерта, здатного достовірно передати первинні оцінки сторінки сайту. В цілому, нейронна мережа та її результати носять виключно дослідницький та ознайомлювальний характер і не пристосовані для автоматизованої обробки зображень.

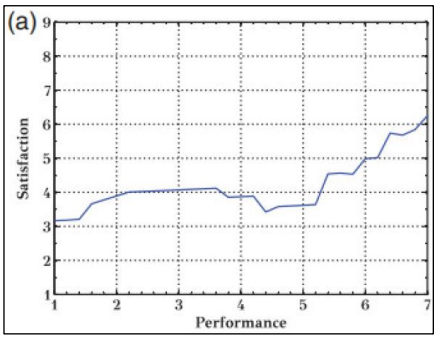


Рисунок 1.4 – Співвідношення продуктивності до задоволеності метрики understability

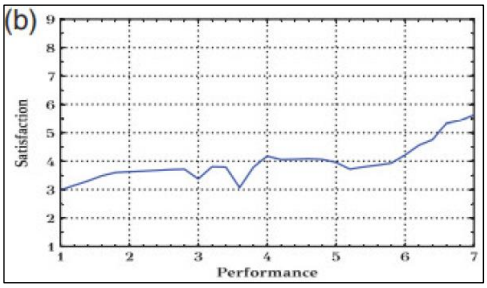


Рисунок 1.5 – Співвідношення продуктивності до задоволеності метрики well-presented

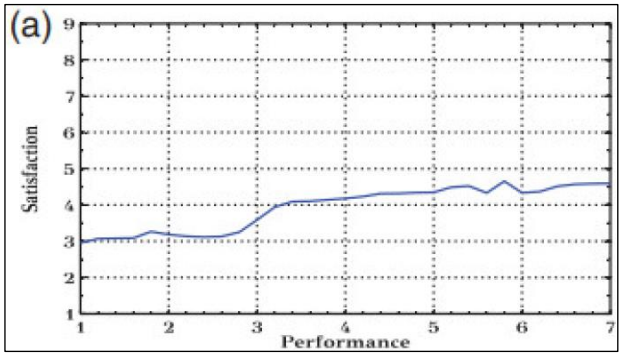


Рисунок 1.6 – Співвідношення продуктивності до задоволеності метрики well-organized

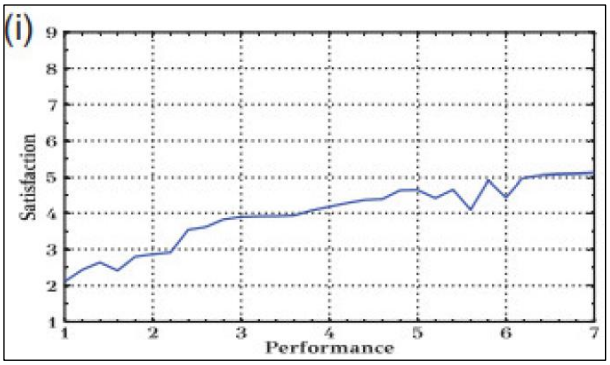


Рисунок 1.7 – Співвідношення продуктивності до задоволеності метрики ease of use

Огляд реалізації пошуку елементів UI з використанням нейронних мереж.

У роботі [22] надається вичерпна інформація про продуктивність моделей нейронних мереж. У порівнянні беруть участь Faster RCNN, Cascade RCNN, YOLOV4 на рисунку 1.8.

Run ID	Model	Overall precision	mAP@IoU 0.5	recall@IoU 0.5
67413	**baseline** Faster RCNN	0.94789	57.2	40.3
67833	Cascade RCNN	0.95035	68.16	53.3
67710	Cascade RCNN	0.94909	64.92	50.5
67722	Cascade RCNN	0.93463	72.33	58.5
67829	YOLOv4	0.93300	73.82	55.6
67707	YOLOv4	0.93125	79.24	59.4
67831	YOLOv4	0.92987	79.11	60
67972	Cascade RCNN	**0.95044**	71.53	55.6
67706	YOLOv4	0.93437	**79.36**	**59.8**

Run 67972: 10000 iterations; Run 67706: 7000 iterations

Рисунок 1.8 – Результати продуктивності нейронних мереж

Як бачимо, YOLOv4 показує хороший результат продуктивності. Варто уточнити, що під час вибору засобів реалізації та технічного експерименту було

вирішено використовувати YOLOv5s, оскільки вона повертає відповідь швидше, що вкрай важливо для серверних додатків.

Для навчання нейронної мережі пошуку елементів UI використовувалося 2950 зображень: 2363 зображень використовувалися для навчання нейронної мережі, 587 зображень – для тестування.

Отримані знання під час аналізу суміжних програм дозволили вибрати засоби реалізації, припустити майбутню архітектуру системи, обсяг необхідних експертних, тренувальних, перевірочних та тестових даних.

1.6 Вибір засобів реалізації

Вибір засобів реалізації заснований на отриманих під час навчання знаннях у галузі нейронних мереж, а також на результатах проектних та курсових робіт.

Була обрана модель нейронної мережі YOLOv5 для класифікації елементів сайту, оскільки вона досить компактна і результативна і відповідає вимогам продуктивності серверних додатків. Відповідно фреймворк для виконання нейронної мережі – PyTorch, мова програмування Python, фреймворк – FastApi. Оскільки деякі компоненти системи були розроблені в ході виконання виробничої практики та курсової роботи, друга нейронна мережа, яка виконує роль експерта, реалізована з використанням фреймворку Keras, який пропонувався для роботи під час навчання. Різнорідність – це один із критеріїв якості інтерфейсу користувача, тому вирішено не використовувати фреймворки для створення SPA додатків, очевидно, що на даному етапі створення клієнта – це зайве.

З тієї ж причини не враховується пошук інструментів і технологій для баз даних через невеликий функціонал програми – користувачеві просто нема за що зберігати зображення для подальшої роботи.

Користувачеві надано можливість завантажити поточний результат, цього цілком достатньо.

Технічний експеримент.

Як фреймворк для реалізації API сервера були відібрані два: Flask і FastApi. Обидва фреймворки позиціонуються як прості для розуміння та легкі у

використанні.

Для вибору оптимального фреймворку був поставлений технічний експеримент: реалізований прототип програми з використанням Flask і FastApi, а як нейронна мережа, що класифікує об'єкти – стандартні передбачені моделі для YOLOv5s та YOLOv3. У результаті було обрано фреймворк FastApi, оскільки має реалізовану асинхронність, можливість легко створювати паралельні потоки, легкими і гнучкими інструментами для налаштування ендпоінтів сервера, наприклад легка і прозора реалізація валідації.

В якості попередньої моделі була обрана YOLOv5 - як більш продуктивна версія і вимагає менший обсяг пам'яті протягом усієї роботи. Важливим фактором стала швидкість навчання, оскільки єдине місце для навчання нейронної мережі надає Google Colab, який має технічні обмеження на ресурси і час, що використовуються. Можна побачити різницю у необхідних ресурсах на рисунках 1.9, 1.10.

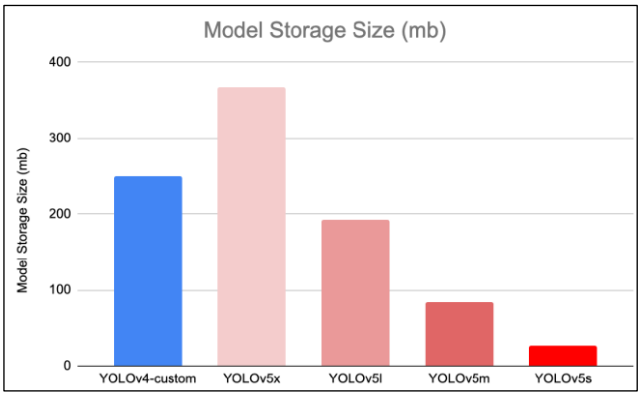


Рисунок 1.9 – Діаграма обсягів файлу моделі у мегабайтах

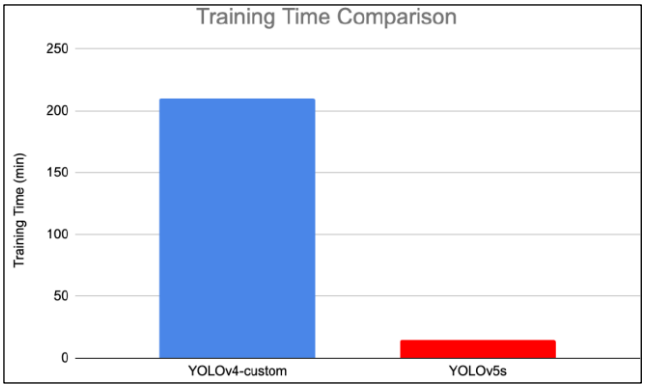


Рисунок 1.10 – Діаграма швидкості навчання нейронної мережі у хвилинах

Вибір спрощеної моделі позитивно позначився якості навченої моделі: більше епох підвищило точність класифікації. В рамках поставленого завдання, в першу чергу, необхідно знайти всі класи, представлені на зображенні.

Точність YOLOv5s менша, ніж у YOLOv3, тому було вирішено запропонувати користувачеві вибрати модель у випадку, якщо швидкість не така важлива. Вибір YOLOv5s пов'язаний також з тим, що при масовій обробці час очікування прийнятний, наприклад, класифікація об'єктів 850 зображень FullHD розміру (1920 на 1080 пікселів) в багатопоточному, асинхронному режимі займає не більше двох хвилин з використанням ЦП AMD FX-6300 6 6 потоків замість 4-5 хвилин з використанням YOLOv3 та повним завантаженням ЦП. Більш детальна інформація наведена в розділі тестування продуктивності системи. На момент вибору інструментів розробки системи поточної інформації достатньо.

2 ПРОЕКТУВАННЯ ВЕБ-ДОДАТКУ ДЛЯ ОЦІНКИ ІЄРАРХІЧНОСТІ ТА РІЗНОРІДНОСТІ ІНТЕРФЕЙСУ ВЕБ-СТОРІНОК ЗА ДОПОМОГОЮ НЕЙРОМЕРЕЖНИХ ТЕХНОЛОГІЙ

2.1 Аналіз вимог

Система оцінки ієрархічності інтерфейсу веб-сторінок, що розробляється, є веб-додатком, який дозволяє, виходячи з метрик, розрахованих на основі попередньо обробленого скріншота веб-сторінки, здійснити їх подальшу обробку за допомогою нейронної мережі і відобразити користувачеві результат оцінки інтерфейсу.

Проектована система буде клієнт-серверним додатком, що забезпечує отримання вихідного зображення – скріншот сторінки сайту, аналіз отриманих даних, надання інформації про якість інтерфейсу з точки зору різнорідності графічних елементів.

Оскільки різнорідність – це один з критеріїв якості інтерфейсу користувача, тому вирішено зробити більше зусилля в розробці розширюваної, продуктивної серверної частини і надати сторінку, що пропонує взаємодію з API і документацію до інтерфейсу.

Для використання програмного забезпечення потрібно пристрій, здатний виконувати байт код Python програм. Вимоги складені, виходячи з поставлених завдань та отриманої теорією.

На рисунку 2.1 зображено діаграму варіантів використання процесу взаємодії користувача з системою. В рамках діаграми актор «Користувач», що є користувачем системи, здійснює взаємодію із системою оцінки ієрархічності інтерфейсу веб-сторінок.

Користувач має можливість завантажити зображення інтерфейсу користувача (варіант використання «Завантажити скріншот веб-сайту»), а також ознайомитися з результатами оцінки завантаженого зображення системою (варіант використання «Отримати результат»). Варіант використання «Отримати результат» також включає варіант використання «Визначити якість інтерфейсу

користувача» де за розрахованими метриками скріншота визначається якість інтерфейсу користувача.

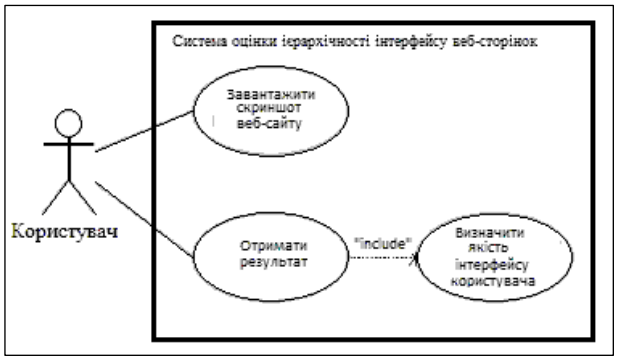


Рисунок 2.1 – Діаграма варіантів використання

У рамках розробки інформаційної системи було виділено такі функціональні вимоги:

- повинне бути забезпечене завантаження графічних зображень за допомогою інтуїтивно зрозумілого інтерфейсу користувача;
- повинна бути забезпечена обробка завантаженого зображення та його подальший аналіз системою;
- система має надавати результат аналізу зображення на основі нормативних показників;
- система має надавати статистичну інформацію на основі проведеного аналізу;
- система має надавати файли з результатами;
- система має надавати можливість інтеграції;
- система має надавати можливість вибору моделі класифікації;
- система повинна надавати користувачеві інтуїтивно зрозумілий інтерфейс користувача для завантаження скріншота веб-інтерфейсу;
- система повинна забезпечити обробку завантаженого зображення та його подальший аналіз на основі одержаних характеристик;
- система повинна відображати результат оцінки веб-інтерфейсу на основі зображення.

У рамках розробки інформаційної системи було виділено такі

нефункціональні вимоги:

– реалізація нейронної мережі, серверна частина повинна бути написана мовою Python з використанням фреймворку FastApi;
– система має бути доступна через браузери Chrome, Firefox, Opera, Microsoft Edge;
– система повинна бути реалізована мовою python;
– у системі мають бути передбачені можливості роботи з різними форматами зображень, такими як .jpg, .png.

2.2 Розробка методу оцінки ієрархічності веб-сторінок

Ієрархічність інтерфейсу веб-сторінок – це ступінь співвідношення площ та взаємопов'язаності елементів інтерфейсу веб-сторінки між собою, а саме: кнопок, тексту, малюнків, посилань та інших елементів.

Так як елементи інтерфейсу користувача можуть мати складну форму, мати відблиски, тінями та іншими декоративними ефектами, будемо розглядати елементи інтерфейсу в якості контурів прямокутної форми.

Контурне представлення елементів інтерфейсу користувача зображено на рисунку 2.2.

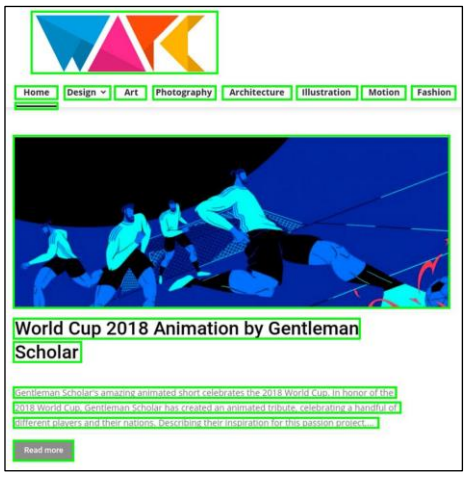


Рисунок 2.2 – Контурне представлення інтерфейсу веб-сторінки

Через відсутність затвердженого методу оцінки ієрархічності інтерфейсу веб-сторінок виникає необхідність розробки власного методу оцінки.

В рамках дослідження був інтерфейс сформований наступний перелік метрик для оцінки ієрархічності:

– наявність вирівнювання елементів у координатній сітці;

– візуальність;

– загальна щільність;

– відносна зв'язність.

Коли елемент вирівняно на координатній сітці, ця функція приймає значення 1 або 0, залежно від вирівнювання елементів інтерфейсу користувача на веб-сторінці.

Метрика розраховується за такою формулою 1:

$$M_1 = \begin{cases} 1 \; x_i = x_j \vee y_i = y_j \\ 0 \; x_i \neq x_j \wedge y_i \neq y_j \end{cases}, \tag{1}$$

де x_i – координата точки контуру елемента інтерфейсу користувача 1 по горизонталі;

x_j – координата точки контуру елемента інтерфейсу користувача 2 по горизонталі;

y_i – координата точки контуру елемента інтерфейсу користувача 1 по вертикалі;

y_j – координата точки контуру елемента інтерфейсу користувача 2 інтерфейсу по вертикалі.

На рисунку 2.3 зображено з наявністю вирівнювання елементів осях координат. Рожевою пунктирною лінією показані лінії вирівнювання кількох елементів.

Рисунок 2.3 – Інтерфейс із наявністю вирівнювання елементів по осях координат

На рисунку 2.4 зображено приклад інтерфейсу з відсутністю вирівнювання елементів осях координат.

Рисунок 2.4 – Інтерфейс із відсутністю вирівнювання елементів по осях координат

Візуальність дозволяє користувачеві досить швидко знайти необхідну інформацію на веб-сторінці, функціональний елемент інтерфейсу або зображення.

Основним параметром, що визначає значення даної метрики є площа елемента.

На рисунку 2.5 зображено інтерфейс веб-сторінки з елементами різної площі. Елемент із зображенням, що володіє найбільшою площею серед усіх елементів, привертає максимальну увагу користувача, як і заголовок, який має досить велику площу по відношенню до інших елементів, що містять текстову інформацію.

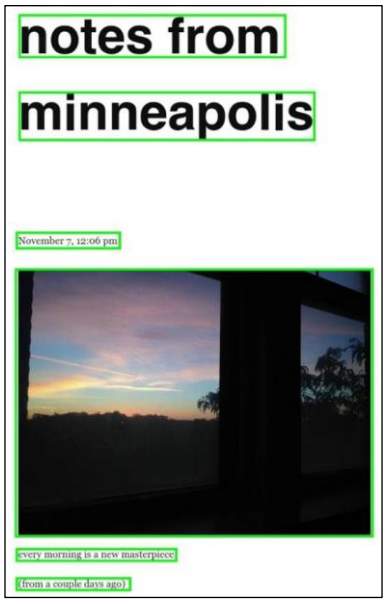


Рисунок 2.5 – Інтерфейс з елементами різної площі

Метрика розраховується за такою формулою 2:

$$M_2 = \frac{1}{n}\sum_{i=1}^{n}\frac{S_i}{S}, \quad (2)$$

де S_i – площа, що займається елементом інтерфейсу користувача на скріншоті інтерфейсу користувача;

S – загальна площа скріншоту інтерфейсу користувача;

n – кількість елементів інтерфейсу користувача.

Загальна щільність дозволяє оцінити перевантаженість інтерфейсу елементами. Перевантаженість ускладнює процес сприйняття інформації користувачем, що, як наслідок, знижує рівень задоволеності інтерфейсом [14].

На рисунку 2.6 зображено скріншот інтерфейсу користувача з великою кількістю елементів, що ускладнює процес взаємодії користувача з веб-сайтом.

Рисунок 2.6 – Перевантажений елементами інтерфейс

На рисунку 2.7 зображено скріншот з невеликою кількістю елементів, що робить процес взаємодії користувача з веб-сайтом простим та очевидним.

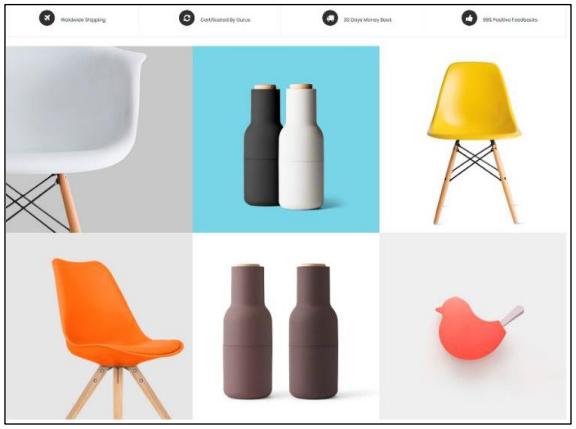


Рисунок 2.7 – Неперевантажений елементами інтерфейс

Метрика розраховується за такою формулою 3:

$$M_3 = \frac{S\ \text{пикс}^2}{n\ \text{шт}}, \tag{3}$$

де S – загальна площа скріншоту інтерфейсу користувача;

n – кількість елементів інтерфейсу користувача.

Нехай $G(V,E)$ – плоский граф, де V – безліч вершин графа, за які прийматимемо безліч елементів веб-інтерфейсу, E – безліч ребер графа, що характеризують зв'язки між елементами веб-інтерфейсу.

На рисунку 2.8 зображено метод побудови графа зв'язності на прикладі однієї вершини.

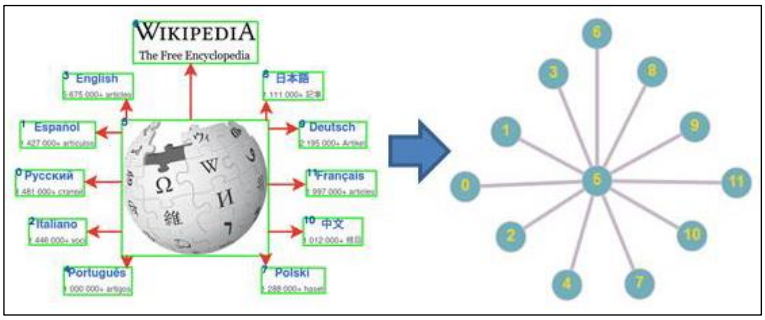


Рисунок 2.8 – Побудова ребер графа зв'язків для вершини

Два елементи веб-інтерфейсу вважатимемо пов'язаними, якщо існує пряма лінія, яка описується рівнянням за формулою 4:

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1}, \quad (4)$$

де x_1 – координата точки контуру елемента інтерфейсу користувача 1 по ширині;

y_1 – координата точки контуру елемента інтерфейсу користувача 1 по висоті;

x_2 – координата точки контуру елемента інтерфейсу користувача 2 по ширині;

y_2 – координата точки контуру елемента інтерфейсу користувача 2 по висоті.

Метрика розраховується за такою формулою 5:

$$M_4 = \frac{k}{n}, \quad (5)$$

де n – кількість елементів інтерфейсу користувача;

k – кількість елементів, для яких дотримується нерівність (формула 6):

$$deg\ (v) > \frac{1}{|V|} \sum_{v \in V} deg\ (v), \quad (6)$$

де $deg(v)$ – ступінь вершини v графа $G(V, E)$;

|V| – кількість вершин графа $G\ (V, E)$.

2.3 Розробка методу оцінки різнорідності веб-сторінок

Метод заснований на використанні нейронних мереж, головним завданням є вибір необхідних і доступних критеріїв для експертної нейронної мережі, а також коректний збір даних для обох нейронних мереж, розбиття даних на навчальну, перевірочну та тестову вибірки, а також достовірна оцінка тестової та перевірочної вибірок у ролі експерта.

Вхідні дані для першої нейронної мережі – це зображення. Мета нейронної мережі – знайти об'єкти. Розмічений датасет містить 8 класів:

- кнопка;
- поле введення;
- спливаюче вікно;

– заголовок;
– зображення;
– підпис, тег;
– посилання;
– текст.

Це вважатимуться виходом нейронної мережі.

Нейронна мережа, що є експертом, теоретично може працювати з кількісними даними: кількість класів, загальна кількість об'єктів, кількість кожного об'єкта класу. У результаті є 10 метрик для експертної нейронної мережі. Вихідним значенням має стати булева змінна - чи є UI різнорідним чи ні.

2.4 Алгоритм роботи та поведінкові аспекти системи

Алгоритм роботи системи містить такі етапи:

– підготувати зображення;
– розрахувати метрики у відповідність до розробленого методу оцінки;
– навчити нейронну мережу;
– отримати оцінку якості за допомогою навченої нейронної мережі.

Етап 3 вищеописаного алгоритму виконується одноразово. Далі обробка розрахованих значень метрик здійснюється за допомогою навченої нейронної мережі.

На рисунку 2.9 зображено діаграму діяльності процесу обробки файлу зображення.

Першим кроком здійснюється розрахунок метрики для завантаженого файлу «Отримати метрики з файлу». Далі здійснюється запуск нейронної мережі «Запустити нейронну мережу» та отримання результатів за результатом її роботи «Отримати результати». У разі відповідності отриманих результатів класу інтерфейсів з хорошим юзабіліті користувач отримує повідомлення про високу оцінку юзабіліті інтерфейсу користувача «Показати хорошу оцінку юзабіліті», в іншому випадку користувач отримує повідомлення про низьку оцінку

користувальницького інтерфейсу «Показати погану оцінку.

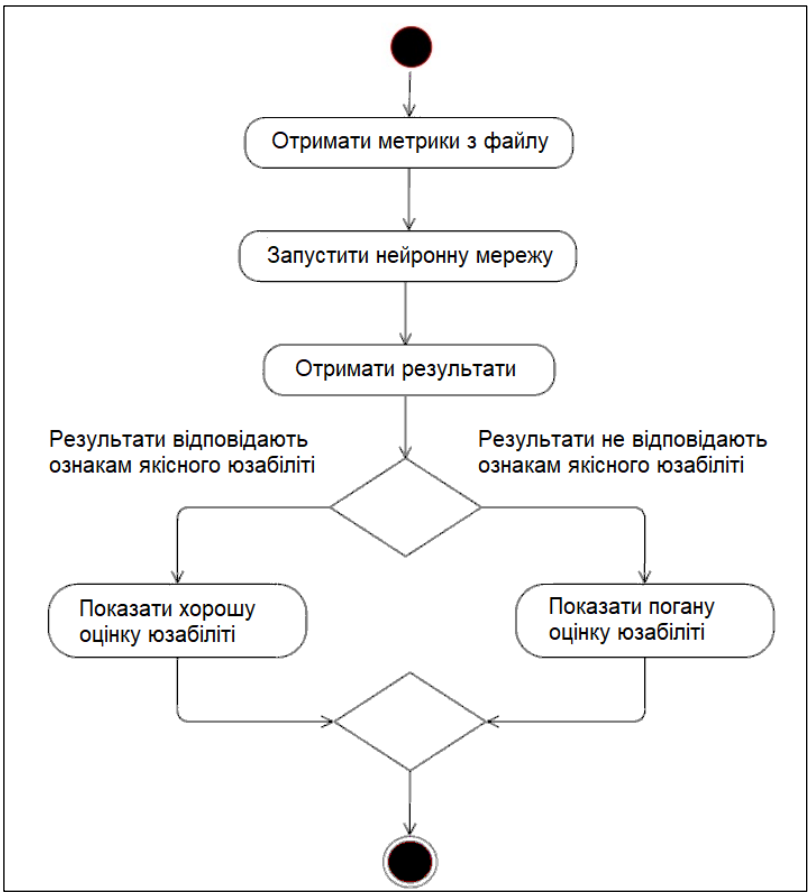


Рисунок 2.9 – Діаграма діяльності процесу обробки файла

2.5 Архітектура системи

Архітектура проектованої системи базується на моделі клієнт-серверної архітектури. Інтерфейс взаємодії користувача з системою є веб-сторінкою, на якій міститься форма для завантаження зображення, а також область для відображення результатів обробки зображення та повідомлень про помилки.

Діаграма розміщення інформаційної системи зображено на рисунку 2.10.

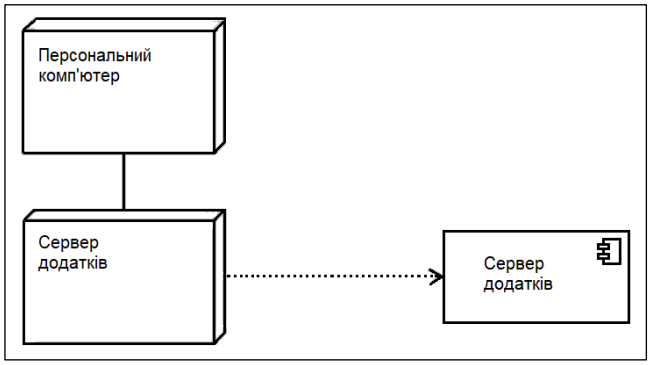


Рисунок 2.10 – Діаграма розміщення

Користувач взаємодіє через свій персональний комп'ютер із сервером програм «Сервер додатків», на якому розміщено веб-проект, що забезпечує функціонал для завантаження та подальшої обробки завантаженого зображення у системі. Після обробки зображення користувач отримує результат як відповідь на HTTP-запит в інтернет-браузері.

В рамках етапу проектування розроблено метод оцінки ієрархічності інтерфейсу веб-сторінок, визначено етапи алгоритму роботи системи, спроектовано поведінкові аспекти системи та архітектуру програми.

Різнорідність елементів GUI – це один із критеріїв оцінки якості, звідси передбачається концепція приватних-публічних методів: критерії оцінки доступні користувачеві, внутрішня реалізація, складові цієї оцінки обчислюються всередині системи незалежно від дій користувача. Таке уявлення дозволить розширювати функціонал, зробити систему гнучкою до модифікації, покращення.

Алгоритм роботи системи представлений на рисунку2.11.

Система приймає на вхід зображення сторінок, завантажених користувачем. Потім система готує зображення для нейронної мережі: нейронна мережа працює тільки з RGB зображеннями, без альфа-каналу, розміри зображення залишаються як є. Підготовлені зображення передаються нейронної мережі – класифікатору для пошуку об'єктів та класифікації елементів веб-сторінки.

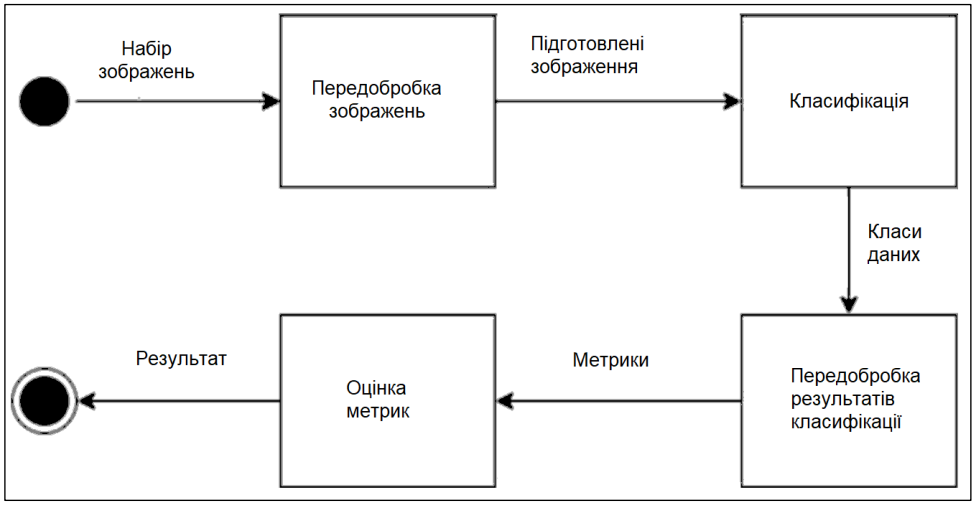


Рисунок 2.11 – Алгоритм роботи системи

Масив класифікованих об'єктів та інформація про них обробляються системою

передачі експертної нейронної мережі. Отримані значення нормалізуються та стають метриками експертної нейронної мережі. Експертна нейронна мережа на основі отриманих метрик (10 значень) видає оцінку від 0 до 1. Система на основі отриманого числа видає текстове повідомлення, разом із зображенням та таблицею знайдених об'єктів. Текстове повідомлення системи ґрунтується на оцінці, де межа різнорідного інтерфейсу все, що нижче або дорівнює 0.65 і не різнорідним, що вище за граничне значення. Це значення було отримано експериментально на тестовій вибірці.

Архітектуру системи найпростіше пояснити на діаграмі компонентів. Передбачається реалізація клієнт-серверної програми. Сервер відповідає за надання клієнту публічних методів розрахунку якості нейронної мережі. Сервер повертає HTML сторінки на запити, отримані від браузера користувача. На основі діаграми use-case буде реалізовано програму з можливістю завантажити та обробити (отримати інформацію) від сервера про якість UI з погляду різнорідності системи.

На рисунку 2.12 представлено діаграму компонентів.

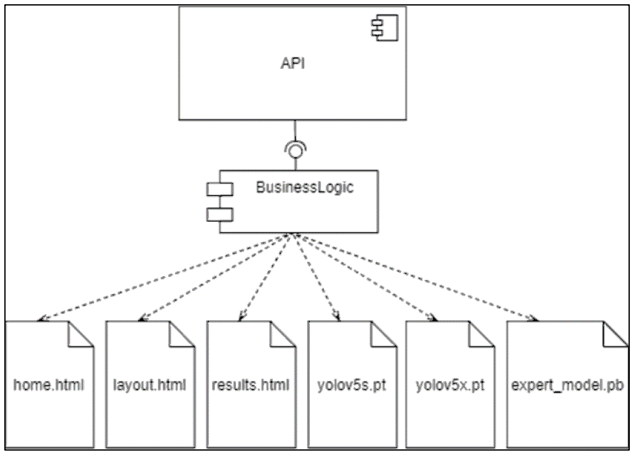


Рисунок 2.12 – Діаграма компонентів

Система складається із компонентів API, BusinessLogic. Взаємодія між компонентами відбувається за допомогою мережного з'єднання між сервером та за допомогою API – публічного інтерфейсу системи. BusinessLogic надає людино-машинний інтерфейс для взаємодії із системою, повертаючи користувачеві html сторінки, ґрунтуючись на шаблонах home.html, layout.html, results.html. Дозволяє

завантажити файл та переглянути результати аналізу, отримані з боку сервера. Надає методи для передачі файлу та отримання результату, відповідає за оцінку інтерфейсу користувача.

Архітектура монолітна, у зв'язку з єдиним обов'язком системи, малою складністю, що проектується. Також через простоту рішення, сервер може бути розміщений у контейнерній віртуалізації та працювати незалежно: для зовнішнього взаємозв'язку потрібен лише один відкритий порт для підключення HTTPS. Шаблони та документація, що надаються разом із сервером, дозволяють швидко налагодити налаштувати, підтримувати у разі розвитку проекту.

На рисунку 2.13 представлено діаграму класів. Діаграма класів надає інформацію про плановані до реалізації класи, не включаючи сторонні класи.

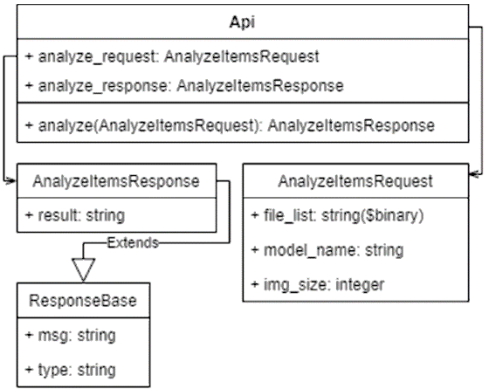


Рисунок 2.13 – Діаграма класів

Клас Api являє собою основний клас для публічної взаємодії з системою, що реалізує публічний метод analyze, виконує класифікацію та експертну оцінку, що використовує нейронну мережу пошуку об'єктів на переданих зображеннях, а також параметри користувача: вибір моделі і розмір зображення, повертає серіалізований JSON рядок з знайденими об'єктами, що містить: ідентифікатор класу, ім'я класу, межі об'єкта (4 точки), достовірність класифікації у вигляді числа з точкою, що плаває.

Обидва методи доступні публічно та реалізують декоратори FastApi, що дозволяють звертатися до них, використовуючи описані HTTP запити та чекати на описаний результат. ResponseBase також служить класом для повернення інформації у разі виникнення помилок валідації або HTTP запитів.

На рисунку 2.14 представлено діаграму варіантів використання.

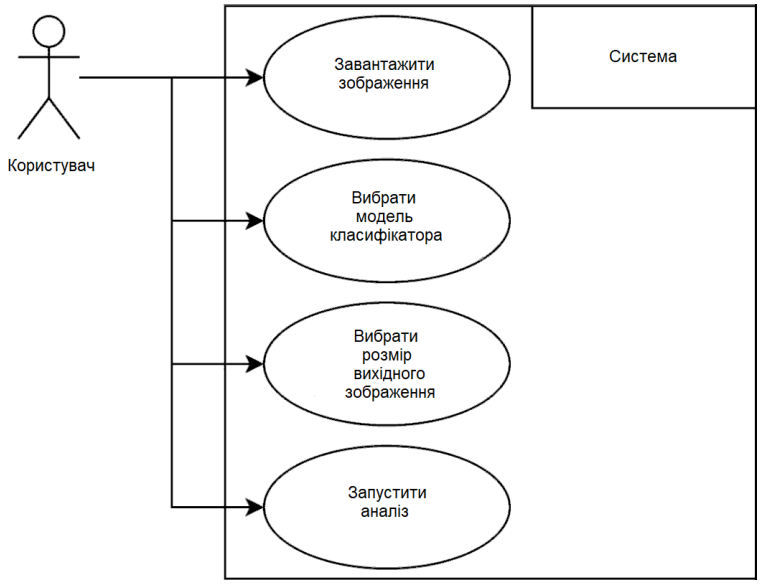


Рисунок 2.14 – Діаграма варіантів використання

Користувач – людина, яка взаємодіє із системою. Користувач може завантажити зображення, інформацію про яку хоче отримати. Інформацією є оцінки елементів, виявлених нейронною мережею на завантаженому зображенні та оцінених нею ґрунтуючись на метриках системи. Користувач може запустити аналіз інтерфейсу користувача, зображеного на завантаженому раніше файлі. Є основною дією, що надає основну логіку системи.

На рисунку 2.15 представлено діаграму станів системи.

Початковий стан системи – очікування (Idle) – система чекає на дію користувача: додавання файлу. Після додавання файлу відбувається перехід у стан перевірки файлу вимогам системи: Validating. При успішній перевірці відбувається завантаження файлу на сервер, інакше система переходить у стан Failed і переходить у початковий стан (Idle) з повідомленням про помилку для користувача. При неможливості відправити файл серверу за будь-якими мережевими помилками (HTTP коди, що передбачають проблеми зі з'єднанням із сервером 4** і 5**) відбувається перехід у стан Failed і виводиться повідомлення про помилку, що відповідає коду помилки.

Успішно отриманий запит обробляється у такому стані – Processing. Далі при порівнянні метрик з отриманими результатами (перевірка якості,

достовірності оцінки) виводиться повідомлення з отриманими результатами (стан Preview Result) або повідомлення про недостатню якість отриманого результату (стан Failed).

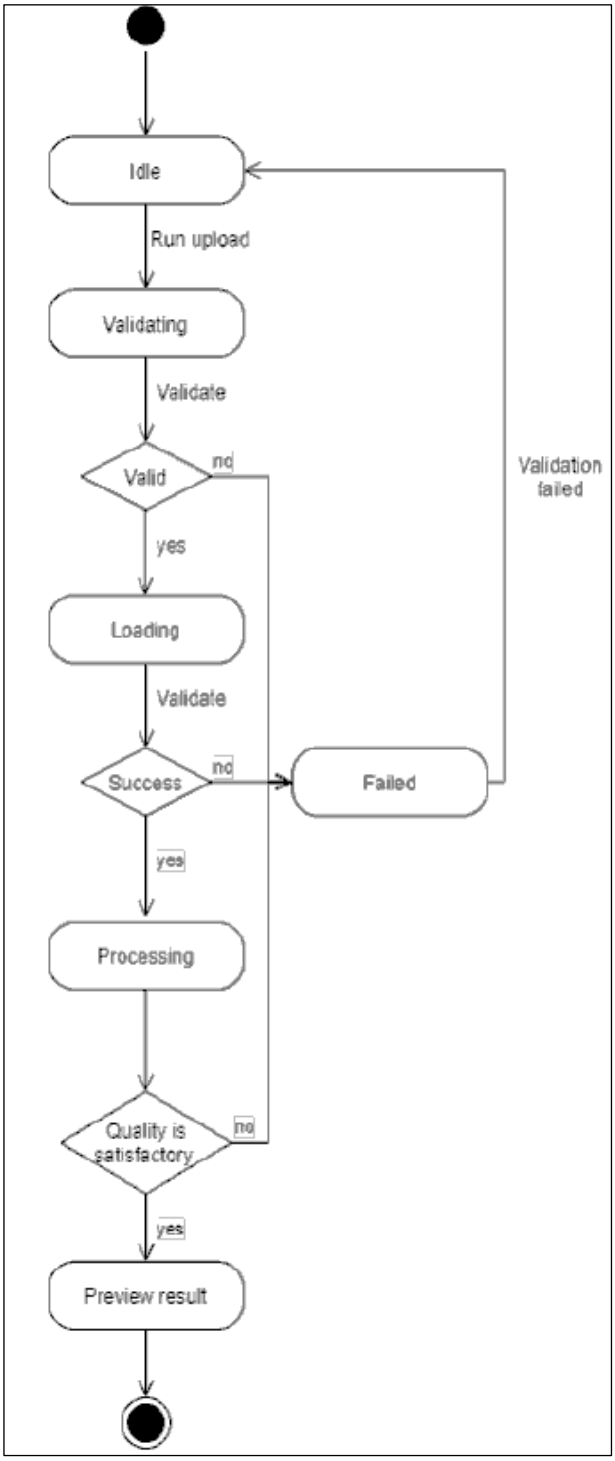


Рисунок 2.15 – Діаграма станів

На рисунку 2.16 представлено діаграму активності.

Upload – операція завантаження зображення. Validate – операція перевірки

завантаженого зображення. Якщо операція закінчилася з негативним результатом, активується стан Failed і відбувається дія ShowErrorReason. Validate для користувача є частиною операції Upload та прихована для нього. Run analysis – основна дія системи, аналіз отриманого зображення нейронною мережею. Compare with metrics – порівняння отриманих даних із метриками системи. Show result – операція подання достовірних результатів аналізу. Show error reason – дія, яка викликається станом Failed при помилках та недостовірних результатах системи. Повідомляє користувачеві про помилку.

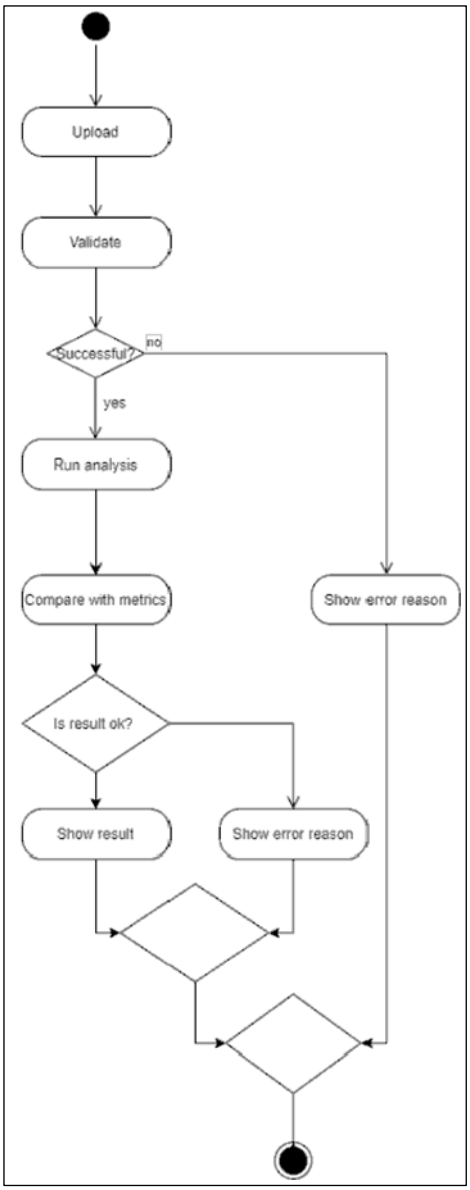


Рисунок 2.16 – Діаграма активності

Спроектована система вийшла простою та зрозумілою. Таке планування

дозволило акцентувати увагу на навчанні моделей нейронних мереж, як результат. Архітектура системи відноситься до монолітного типу, однак, через єдиний обов'язок системи цей проект може стати частиною мікросервісної реалізації.

Надалі такий підхід виправдав себе: виникли труднощі з навчанням експертної нейронної мережі, був потрібний аналіз проблем, повторне анкетування, збільшення даних для навчання.

3 РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ ДЛЯ ОЦІНКИ ІЄРАРХІЧНОСТІ ТА РІЗНОРІДНОСТІ ІНТЕРФЕЙСУ ВЕБ-СТОРІНОК ЗА ДОПОМОГОЮ НЕЙРОМЕРЕЖНИХ ТЕХНОЛОГІЙ

3.1 Підготовка вихідних даних

Для реалізації веб-програми використовувалися такі технології:

- мова програмування python версії 3.12.2;
- бібліотека комп'ютерного зору OpenCV виділення контурів елементів інтерфейсу;
- нейромережні бібліотеки Keras та Tensorflow для розробки та навчання нейронної мережі;
- фреймворк Flask для реалізації веб-програми мовою Python.

Як основний набір даних було обрано набір даних «Web Pages» з сайту https://www.kaggle.com/datasets. У цьому наборі міститься 1500 скріншотів веб-сайтів різної спрямованості: порталів новин, пошукових систем, інтернет-магазинів, відео стрімінгових сервісів, блогів, текстових сайтів, сайтів, присвячених музиці, спорту, приготуванню їжі та інших.

Далі з основного набору даних було виділено 100 скріншотів, що відповідали певним раніше метрикам для оцінки ієрархічності: наявності вирівнювання елементів координатної сітці, візуальності, загальної щільності. Відносна зв'язність не включалася до переліку метрик через трудомісткість візуальної оцінки. Поєднання метрик наведено у таблиці 3.1.

Таблиця 3.1 – Поєднання метрик на формування навчальної вибірки

	Поєднання							
Наявність вирівнювання елементів у координатній сітці (1 – є/ 0 – ні)	0	0	0	0	1	1	1	1
Візуальність (1 – великий розмір елемента/ 0 – невеликий розмір елемента)	0	0	1	1	0	0	1	1
Загальна щільність (1 – висока щільність елементів / 0 – низька щільність елементів)	0	1	0	1	0	1	0	1

У ході збору даних та розмітки невеликого обсягу зображень для навчання нейронної мережі були вироблені типи ресурсів (сторінок) за своїм змістом, які до того ж мають характерні візуальні ознаки:

– ынформаційно-довідкові сайти;
– блоги, соціальні мережі;
– сайт візитівка, сайт виробника, портфоліо;
– сайт пошуку роботи, оголошень, каталоги, інтернет-магазини;
– відео-фото-хостинги, музичні та кіносервіси;
– форуми: сайти завантажень, торрент-трекери, форуми;
– портали.

Не менш важливо визначити такі типи вмісту:

а) динамічні:
   1) відео;
   2) анімація;
б) інтерактивні:
   1) кнопки;
   2) поля введення;
   3) гіпертекст;
в) статичні:
   1) текст;
   2) зображення.

Використані датасети.

У ході пошуку зібраних зображень сторінок було знайдено два датасети [23], що містять 2700 та 1203 зображень відповідно.

Датасет, наданий сервісом Roboflow, вже розмічений і його можна використовувати для навчання безпосередньо. Він був знайдений пізніше, коли використовувався датасет розміщений у сервісі Kaggle [24].

Для розмітки використовувався сервіс coco-annotator. Сервіс розмітки було розгорнуто локально з використанням docker-compose. На рисунку 3.1 можна побачити перелік контейнерів.

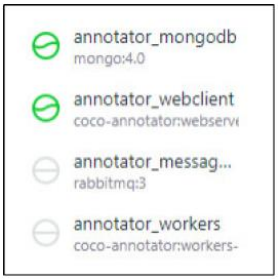


Рисунок 3.1 – Список контейнерів програми інструкції

Coco-annotator дозволяє проводити розмітку об'єктів примітивами та складними фігурами. В рамках мого завдання використовувалися геометричні примітиви для розмітки.

Нижче наведено приклад результатів розмітки об'єктів класифікації (рисунок 3.2).

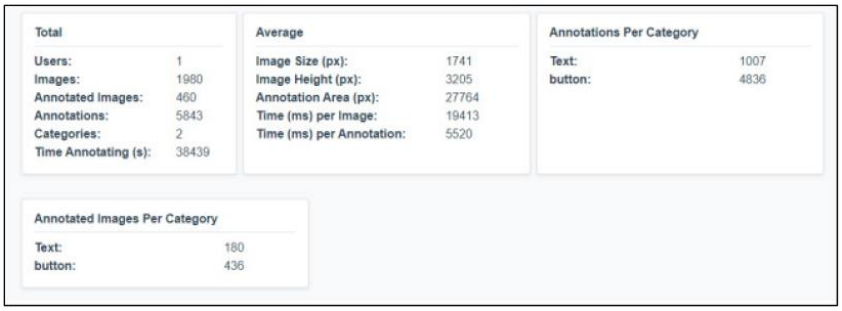


Рисунок 3.2 – Процес розмітки об'єктів класифікації

Підсумком розмітки стало 460 зображень, де в кожному, в середньому містилося від 6 до 15 елементів, що цікавлять. На рисунку 3.3 зображено результат експорту: файл json з інформацією про розмітку.

Рисунок 3.3 – Результат експорту

Файл містить у собі дані про виділення: наданий клас, координати та інша метаінформація.

На рисунку 3.4 зображено деякі категорії (класи).

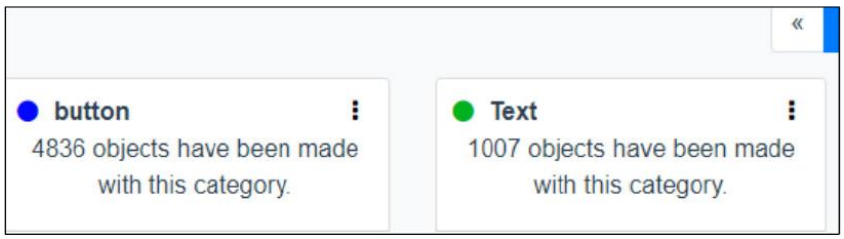


Рисунок 3.4 – Приклад класів та метаінформації про клас

Проблеми з використанням розмічених даних.

Отриманий файл не вдалося використати у подальшій роботі: виникли труднощі з конвертацією COCO JSON файлу з анотацією та зображень у YOLO txt. У процесі роботи з проектом не вдалося приділити достатньо часу для вирішення проблеми конвертації. Тому було вирішено розмітити, використовуючи веб-сервіс Roboflow і надалі дотримуватись їх інструкцій у процесі виконання подальшої роботи.

За основу було взято колекцію розмічених зображень, надану на сервісі Roboflow [23] та розширено власною розміткою деякого числа з вихідного, нерозміченого датасету використовуючи наданий Roboflow сервісом.

Приклад розмітки через сервіс Roboflow наведений на рисунку 3.5.

Рисунок 3.5 – Результат розмітки через сервіс Roboflow на прикладі однієї зі сторінок

Створення даних для експертної нейронної мережі.

У процесі створення набору даних для нейронної мережі було складено два датасети. Для першого датасета було опитано трьох осіб, які безпосередньо займалися завданням якості веб-інтерфейсу. Обсяг датасета 100 зображень пройдено трьома критерієм оцінки. Принцип оцінювання простий: 1 – подобається, 0 – не подобається. Фрагмент результатів представлений на рисунку 3.6. Другий датасет було складено особисто. Обсяг даних було збільшено до 500 зображень, було складено 5 питань. Короткі формулювання та приклад даних зображені на рисунку 3.7.

# картинки	Артем	Марія	Ольга	ОЦІНКА
1	1	1	1	1
2	0	1	1	1
3	0	1	1	1
4	0	0	1	0
5	0	0	1	0
6	0	1	1	1
7	0	1	1	1
8	1	1	0	1
9	0	0	1	0
10	1	1	1	1

Рисунок 3.6 – Фрагмент набору даних

Сценарій використання очевидний	Загальна перевантаженість сторінки	Організація сторінки	Інформація легко сприймається	Розрізненість класів елементів	Оцінка
5	1	5	5	1	0,68
5	2	4	5	1	0,68
3	4	1	2	4	0,56
4	3	3	3	4	0,68
3	2	3	4	2	0,56

Рисунок 3.7 – Фрагмент набору даних, фінальна версія

Для роботи використовувалася лише послідовність оцінок як масив результатів нейронної мережі для навчання.

3.2 Підготовка зображення

Для первинної обробки зображень використовувалася бібліотека комп'ютерного зору OpenCV.

Процес обробки зображення складається з наступних етапів:

– переведення в градацію сірого;

– бінаризація зображення;

– виділення контурів;

– видалення зайвих контурів.

На першому етапі обробки зображення переводилося в градацію сірого кольору, представлене на рисунку 3.8.

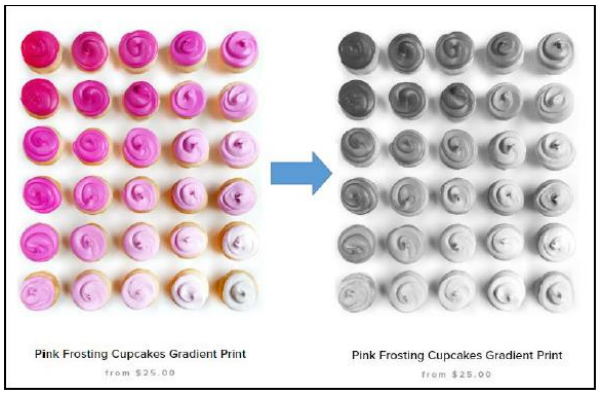


Рисунок 3.8 – Переведення у градацію сірого

Далі здійснювалася бінаризація зображення для отримання зображення чорно-білій гамі, щоб полегшити процес подальшого виділення контурів. На рисунку 3.9 показано результат бінаризації зображення.

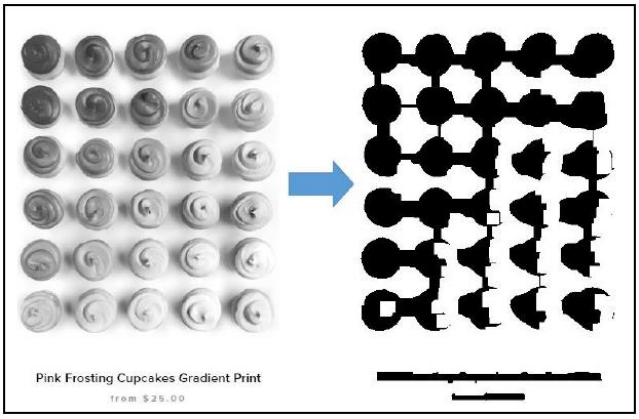


Рисунок 3.9 – Бінаризація

Наступним кроком здійснювалося виділення контурів зображення. Функція виділення контурів зображення показана на лістингу Б.1 додатку Б. Результат виділення контурів зображено на рисунку 3.10.

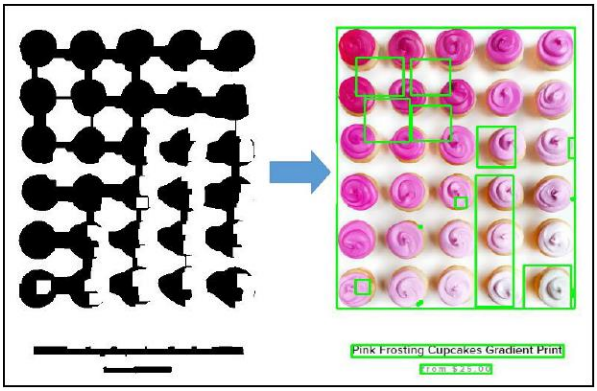


Рисунок 3.10 – Виділення контурів

В результаті первинного розпізнавання, зображеного на рисунку 3.10, алгоритм комп'ютерного зору виявив контури елементів зображення, що включають зайві контури, які не потрібні для подальшої роботи. Після видалення зайвих контурів був отриманий результат, зображений на рисунку 3.11, на зображенні присутні лише контури елементів, які будуть використані для розрахунку метрик.

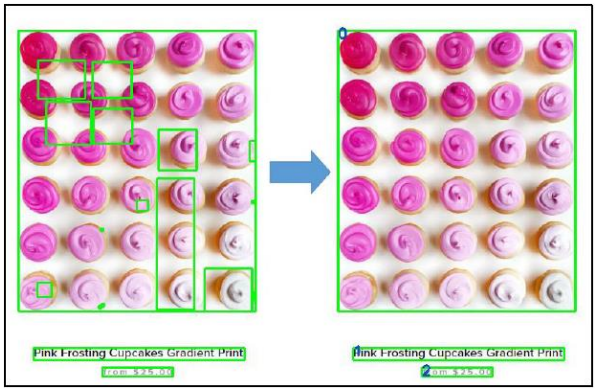


Рисунок 3.11 – Видалення зайвих контурів

3.3 Розрахунок метрик

Відповідно до описаної вище методології оцінки ієрархічної структури веб-інтерфейсу, для кожного скріншоту, включеного до навчального набору, було розраховано чотири показники: вирівнювання на координатній сітці, видимість, загальна щільність і відносна зв'язність. На лістингу Б.2 додатку Б показана реалізація функції розрахунку метрик загальної густини та відносної зв'язності.

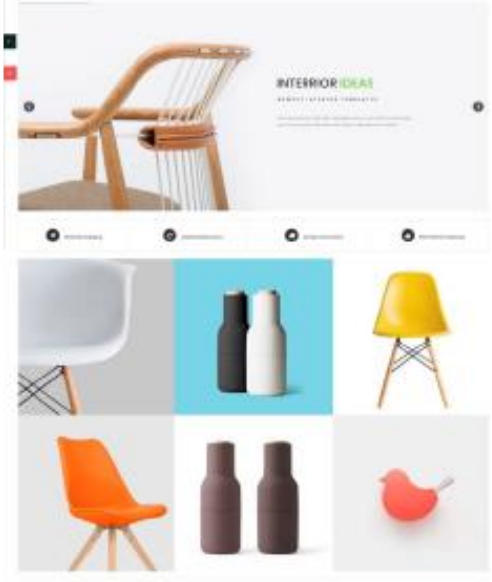
У таблиці 3.2 міститься приклад розрахунку метрик для двох зображень, що

належать до різних груп, принцип формування яких описано раніше.

Перше зображення відповідало групі поєднання метрик: наявність вирівнювання елементів координатної сітці (1), візуальність (0), загальна щільність (1).

Друге зображення відповідало групі поєднання метрик: наявність вирівнювання елементів координатної сітці (1), візуальність (1), загальна щільність (1).

Таблиця 3.2 – Результати розрахунку метрик для двох зображень

Зображення	**Значення метрик**
	Вирівнювання: 1 Візуальність: 0,0005 Щільність: 10 763,2 Відносна зв'язність: 0,442
	Вирівнювання: 1 Візуальність: 0,26 Щільність: 171 498 Відносна зв'язність: 0,446

3.4 Реалізація та навчання нейронних мереж

Для навчального набору даних була сформована вибірка зі 100 скріншотів інтерфейсів користувача сайтів різної спрямованості, принцип підготовки яких був описаний раніше.

Для перевірочного набору даних була сформована вибірка з 20 скріншотів інтерфейсів користувача сайтів різної спрямованості, підібраних аналогічно у відповідність з раніше описаним принципом підготовки даних.

Формування експертної оцінки.

Для оцінювання ієрархічності інтерфейсу веб-сторінок був розроблений опитувальник, що дозволяє експерту дати оцінку інтерфейсу відповідно до виділених в рамках дослідження метриків: наявністю вирівнювання елементів у координатній сітці, візуальністю, загальною щільністю. Відносна зв'язність не включалася до опитувальнику через труднощі візуальної оцінки.

Як шкала для оцінювання перших трьох питань була використана шкала Лайкерта, яка дозволяє визначити ступінь задоволеності, пропонуючи опитуваному вибрати варіант відповіді серед переліку: повністю згоден, частково згоден, важко відповісти, частково не згоден, зовсім не згоден.

Опитувальник містить такі запитання:

- Чи можете Ви легко відрізнити головне від другорядного на даній веб-сторінці?
- Чи можете Ви швидко знайти необхідну інформацію на даній веб-сторінці?
- Чи здається Вам веб-сторінка візуально перевантаженим?
- Вам подобається інтерфейс цієї веб-сторінки?

Відповіді перші три питання планується використовувати у перспективі подальшого дослідження призначення ваг окремих метрик під час навчання нейронної мережі.

У рамках проведення дослідження експертам було поставлено одне питання, яке міститься у опитувальнику за номером 4.

Експерту було запропоновано поставити зображення оцінку 1, якщо йому подобається інтерфейс веб-сторінки та оцінку 0, якщо йому не подобається інтерфейс веб-сторінки.

Оцінка здійснювалася трьома експертами, підсумкова оцінка формувалася за принципом підрахунку середнього арифметичного оцінок зображення із округленням у цілу частину. Фрагмент протоколу оцінки наведено на рисунку 3.12.

# картинки	Артем	Марія	Ольга	ОЦІНКА
1	1	1	1	1
2	0	1	1	1
3	0	1	1	1
4	0	0	1	0
5	0	0	1	0
6	0	1	1	1
7	0	1	1	1
8	1	1	0	1
9	0	0	1	0

Рисунок 3.12 – Експертна оцінка зображень

Реалізація та підбір архітектури нейронної мережі.

При реалізації нейронної мережі за допомогою бібліотек Keras та Tensorflow використовувалася послідовна модель. Як функцію активації внутрішніх шарів було обрано функцію Relu, як функцію активації нейрона останнього шару було обрано сигмоїдальну функцію для прогнозування ймовірнісної оцінки від 0 до 1.

У процесі реалізації було досліджено низку архітектур нейронних мереж з різною кількістю внутрішніх шарів та зв'язків між нейронами внутрішніх шарів.

Перший варіант архітектури нейронної мережі містив два внутрішні шари, 4 нейрони в першому внутрішньому шарі, 8 нейронів у другому внутрішньому шарі (див. рисунок 3.13).

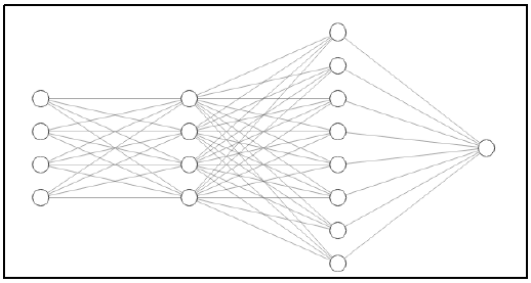


Рисунок 3.13 – Архітектура нейронної мережі. Варіант 1

За результатами навчання на 300 епохах нейронна мережа показала точність 76 % на навчальному наборі даних, 72 % на наборі перевірочного даних.

У другому варіанті архітектури нейронної мережі було збільшено число нейронів другого внутрішнього шару до 12 нейронів у порівнянні з першим варіантом архітектури (див. рисунок 3.14).

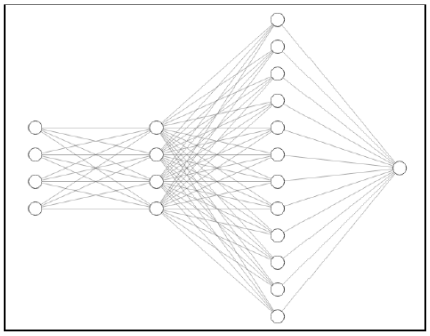


Рисунок 3.14 – Архітектура нейронної мережі. Варіант 2

За результатами навчання на 300 епохах нейронна мережа показала точність 77 % на навчальному наборі даних, 87 % на наборі перевірочному даних.

У третьому варіанті архітектури нейронної мережі було збільшено кількість внутрішніх верств порівняно з попередніми варіантами архітектури. Дана архітектура містила 3 внутрішні шари по 8, 16 і 8 нейронів відповідно (див. рисунок 3.15).

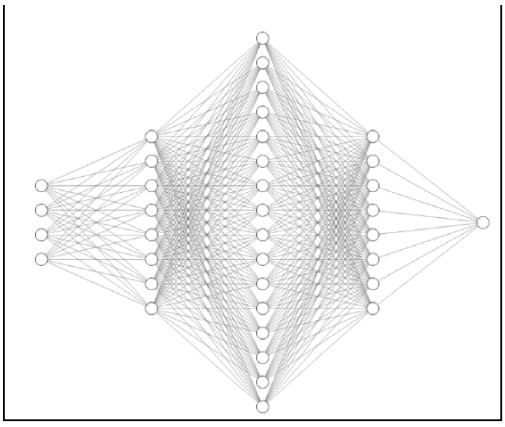


Рисунок 3.15 – Архітектура нейронної мережі. Варіант 3

За результатами навчання на 300 епохах нейронна мережа показала точність 79 % на навчальному наборі даних, 77 % на наборі перевірочному даних.

У четвертому варіанті архітектури нейронної мережі було збільшено кількість

нейронів у внутрішніх шарах, порівняно з попередніми варіантами архітектури. Дана архітектура містила 3 внутрішні шари по 12, 16 і 12 нейронів відповідно (див. рисунок 3.16).

За результатами навчання на 300 епохах нейронна мережа показала точність 79% на навчальному наборі даних, 83% на наборі перевірочного даних.

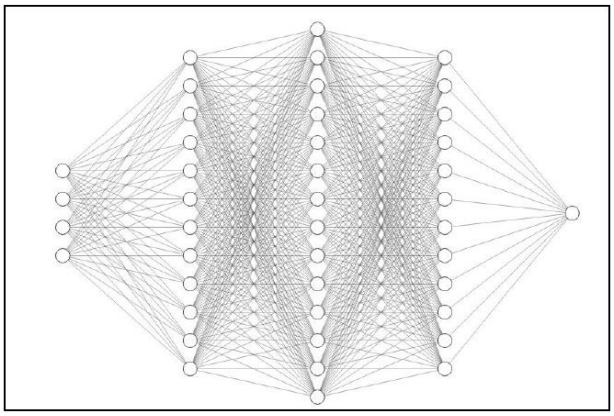


Рисунок 3.16 – Архітектура нейронної мережі. Варіант 4

У таблиці 3.3 наведено зведені результати навчання чотирьох варіантів архітектур нейронних мереж.

Таблиця 3.3 – Результати навчання нейронних мереж різних архітектур

Варіант архітектури нейронної мережі	Точність на навчальному наборі даних	Точність на перевірочному наборі даних
Варіант 1	76%	72%
Варіант 2	77%	87%
Варіант 3	79%	77%
Варіант 4	79%	83%

За результатами порівняння архітектур було ухвалено рішення вибрати Варіант 2 для подальшої роботи через найкращі результати точності розпізнавання на перевірочному наборі даних.

Динаміка змін метрики accuracy для навчального та перевірочного наборів даних для обраної архітектури нейронної мережі наведена на рисунку 3.17.

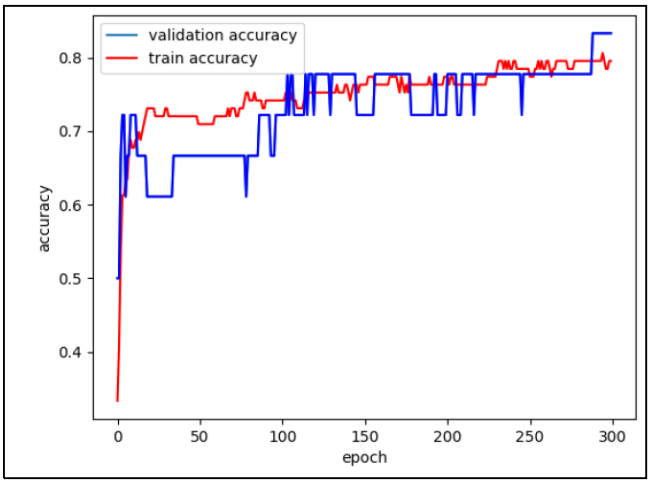


Рисунок 3.17 – Графік зміни метрики accuracy для навчального та перевірочного наборів даних

Динаміка змін значення функції втрат loss для навчального та перевірочного наборів даних наведена на рисунку 3.18.

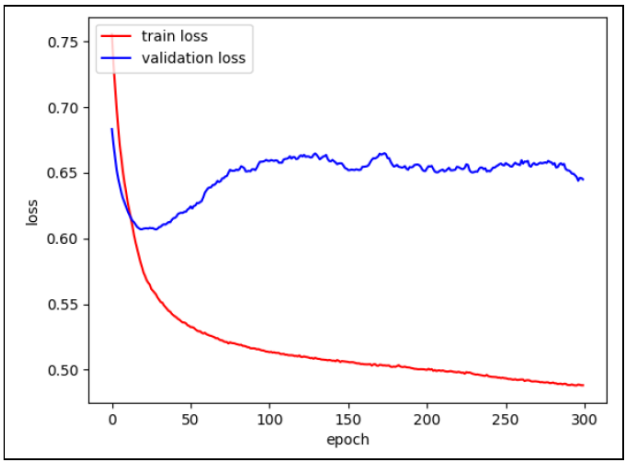


Рисунок 3.18 – Графік зміни значення функції втрат loss для навчального та перевірочного наборів даних

Для реалізації класифікатора веб-елементів нейронної мережі використовувалися передбачені мережі YOLOv5s та YOLOv5x. Нейронна мережа навчена на 1340 зображення різних розмірів і містить різні типи сайтів, розмітка містить 8 класів.

Для навчання використовувалася безкоштовна платформа обчислювальних ресурсів Google Colab. Для навчання використовувалися ресурси графічного прискорювача Nvidia K80.

Використовувався загальнодоступний код навчання нейронної мережі. Були змінені співвідношення пакета (batch) кожного кроку навчання та обсяг тренувальних даних.

Модель нейронної мережі представлена на рисунку 3.19.

Жодних змін у моделі не проводилося, отримані результати, представлені нижче на рисунку 3.20 з діаграмами характеристик нейронної мережі, цілком влаштовують за своєю якістю.

Як можна побачити, навчена модель наблизилася до отриманих результатів розробників YOLOV5. Якість, точність та об'єктивність даних прийнятна.

Далі наведено приклад (див. рисунок 3.21) класифікованого зображення з використанням YOLOV5s.

```
          from  n    params  module                                  arguments
            -1  1      3520  models.common.Focus                     [3, 32, 3]
            -1  1     18560  models.common.Conv                      [32, 64, 3, 2]
            -1  1     19904  models.common.BottleneckCSP             [64, 64, 1]
            -1  1     73984  models.common.Conv                      [64, 128, 3, 2]
            -1  1    161152  models.common.BottleneckCSP             [128, 128, 3]
            -1  1    295424  models.common.Conv                      [128, 256, 3, 2]
            -1  1    641792  models.common.BottleneckCSP             [256, 256, 3]
            -1  1   1180672  models.common.Conv                      [256, 512, 3, 2]
            -1  1    656896  models.common.SPP                       [512, 512, [5, 9, 13]]
            -1  1   1248768  models.common.BottleneckCSP             [512, 512, 1, False]
            -1  1    131584  models.common.Conv                      [512, 256, 1, 1]
            -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
       [-1, 6]  1         0  models.common.Concat                    [1]
            -1  1    378624  models.common.BottleneckCSP             [512, 256, 1, False]
            -1  1     33024  models.common.Conv                      [256, 128, 1, 1]
            -1  1         0  torch.nn.modules.upsampling.Upsample    [None, 2, 'nearest']
       [-1, 4]  1         0  models.common.Concat                    [1]
            -1  1     95104  models.common.BottleneckCSP             [256, 128, 1, False]
            -1  1    147712  models.common.Conv                      [128, 128, 3, 2]
      [-1, 14]  1         0  models.common.Concat                    [1]
            -1  1    313088  models.common.BottleneckCSP             [256, 256, 1, False]
            -1  1    590336  models.common.Conv                      [256, 256, 3, 2]
      [-1, 10]  1         0  models.common.Concat                    [1]
            -1  1   1248768  models.common.BottleneckCSP             [512, 512, 1, False]
  [17, 20, 23]  1     21576  models.yolo.Detect                      [3, [[10, 13, 16, 30, 33, 23]
Summary: 283 layers, 7260488 parameters, 7260488 gradients
```

Рисунок 3.19 – Модель нейронної мережі

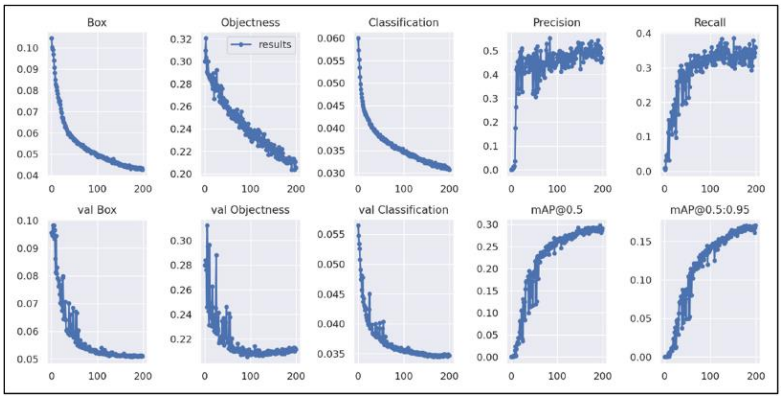


Рисунок 3.20 – Результати навчання нейронної мережі – класифікатора веб-елементів

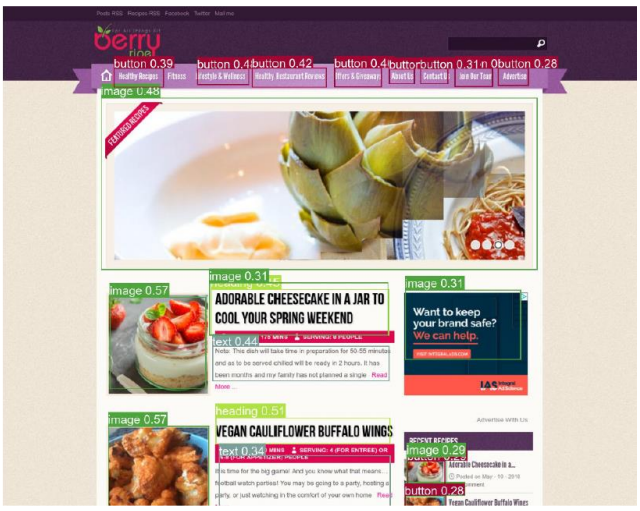


Рисунок 3.21 – Приклад класифікованого зображення

Як можна побачити всі класи, що представлені на сторінці були виявлені. Додавання розмічених зображень більшої роздільної здатності та без стиснення покращили якість класифікації під час реального використання.

Експертна нейронна мережа повністю імплементована самостійно.

Перша реалізація не мала успіху: тестові дані були підготовлені погано, не враховуючи специфіку нейронної мережі, оцінка сторінок також не враховувала ці особливості. Також був очевидним недолік тестових даних. Через війну виникла проблема навчання. Це можна побачити на рисунку 3.22.

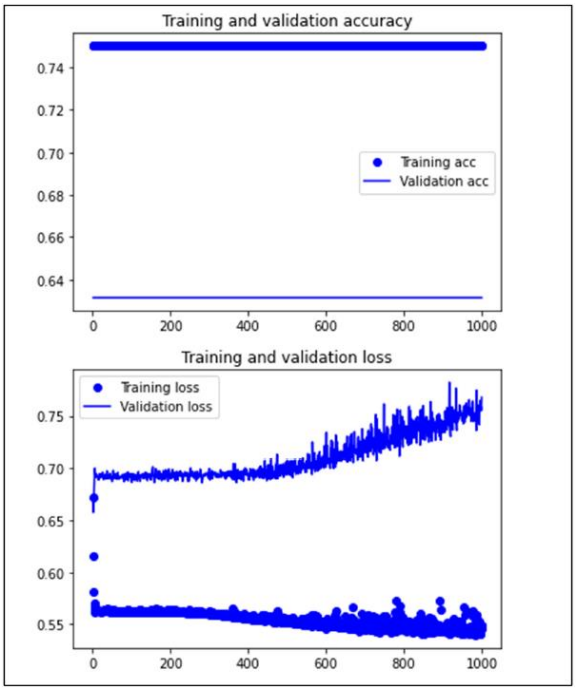


Рисунок 3.22 – Криві точності та втрат першої версії нейронної мережі

З цих графіків можна зрозуміти, що нейронна мережа не може знайти закономірності між вхідними даними та передбачуваною оцінкою. Тому вирішили переоцінити дані.

Разом із зміненими даними нейронна мережа стала давати позитивні закономірності, проте графік на рисунку 3.23 дає зрозуміти, що модель виходить перенавченої, це пов'язані з малої вибіркою.

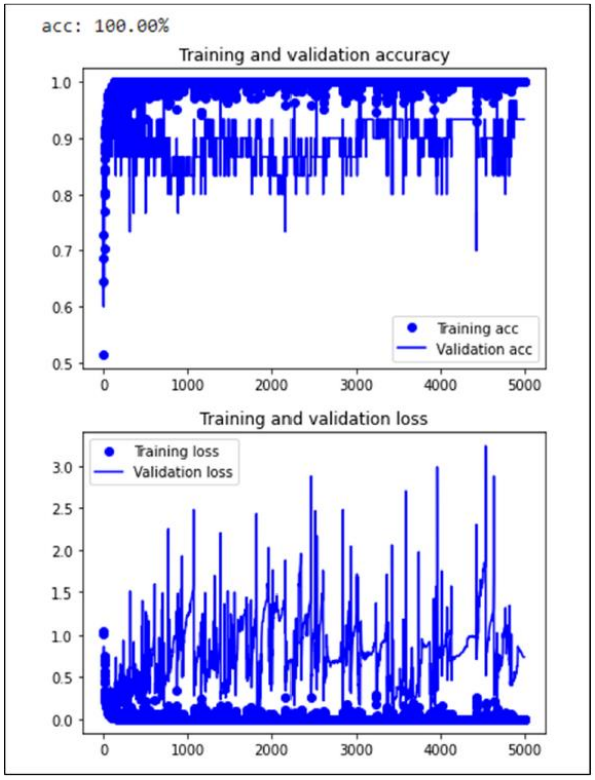


Рисунок 3.23 – Криві точності та втрат першої версії нейронної мережі та зміною тестової вибірки та підходу до неї

Тому було вирішено збільшити їх обсяг із початкових 100 до 500 зображень. Це допомогло навчити нейронну мережу з добрим і достатнім результатом точності. Топологія нейронної мережі представлена на рисунку 3.24.

Модель є багатошаровою нейронною мережею, вхідний шар складається з 10 нейронів з функцією активації ReLU, відповідає числу переданих параметрів, вихідний шар з 1 нейрона з функцією активації ReLU. Внутрішні шари активуються з урахуванням функції ReLU.

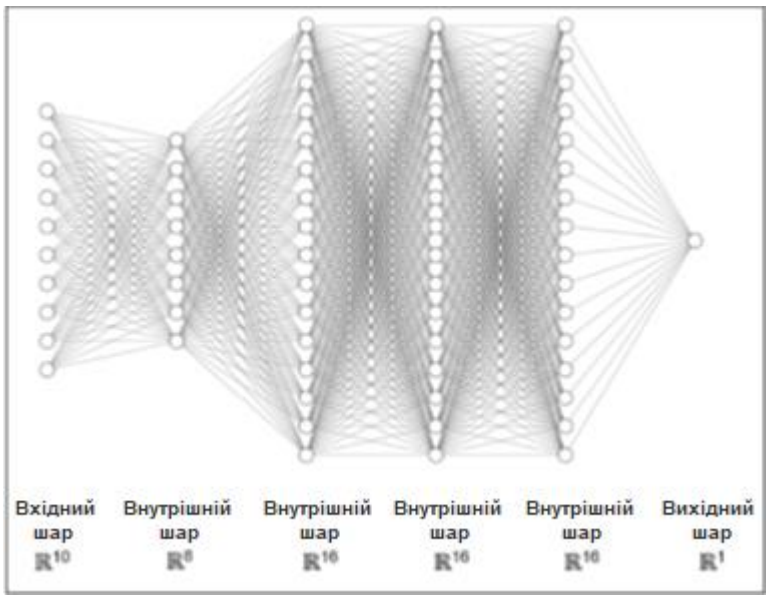


Рисунок 3.24 – Топологія нейронної мережі

Також є шар Dropout (Виняток), для зниження перенавченості нейронної мережі, запобігання складним коадаптаціям нейронів під час навчання на тренувальних даних.

Використовувалась логарифмічна функція втрат, для оптимізації цільової функції використовувався ітераційний метод Adam. Далі представлено реалізацію моделі в лістингу Б.3 додатку Б.

Нижче наведено рисунок 3.25 з прикладом кривої навчання. Один графік визначає точність, інший – значення помилки.

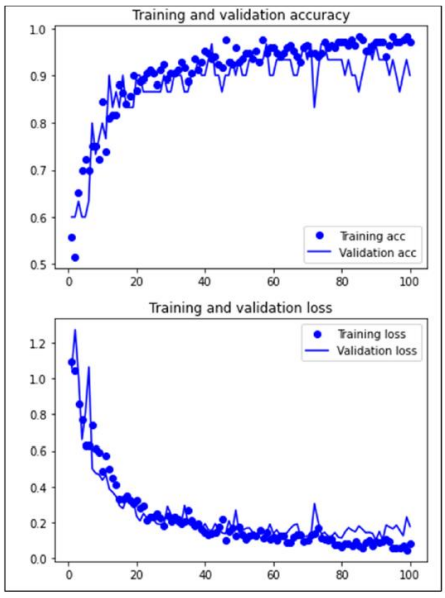


Рисунок 3.25 – Криві точності та втрат першої версії нейронної мережі

Результатом навчання стала нейронна мережа з точністю 92–94 % на тестовій вибірці та 88–90 % на перевірочній вибірці. Перевірка на випадкових незадіяних у процесі навчання зображеннях показала, що точність дорівнює 83%.

У результаті навчання нейронної мережі було отримано безцінний досвід у створення мережі, аналіз проміжних результатів та налагодження.

3.5 Реалізація веб-програми

Реалізація веб-програми здійснена з використанням фреймворку Flask і є веб-сторінкою з розташованою на ній формою завантаження зображення і формою для виведення результатів обробки скріншота і повідомлень про помилки. Інтерфейс сторінки веб-програми зображено на рисунку 3.26.

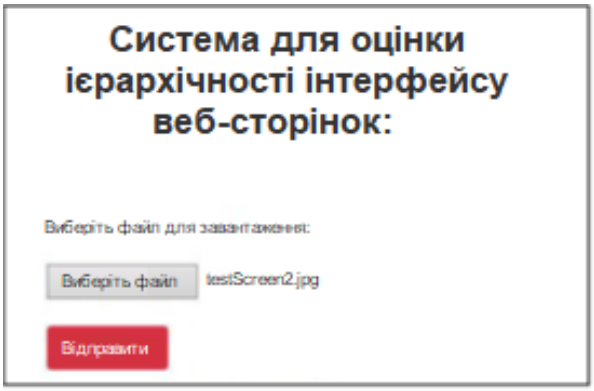


Рисунок 3.26 – Інтерфейс веб-програми

У процесі роботи веб-програми здійснюється завантаження скріншота інтерфейсу веб-сторінки через форму завантаження і після обробки зображення користувач бачить результат, зображений на рисунку 3.27.

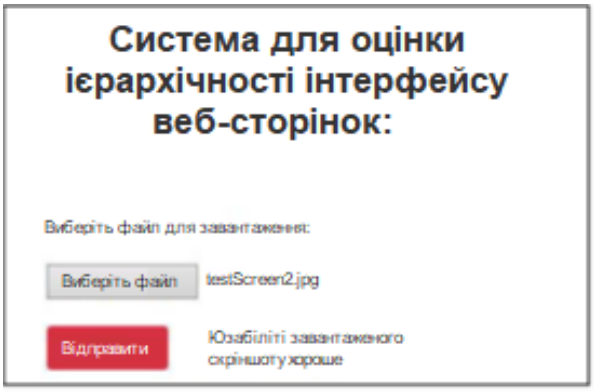


Рисунок 3.27 – Відображення результату перевірки для завантаженого файлу

3.6 Реалізація серверної частини системи

Серверна частина системи виконана на основі фреймворку FastApi.

Нижче наведено лістинг ініціалізації FastApi фреймворку та додавання шаблонів сторінок (див. листинг Б.4 додатку Б).

Імплементований метод повернення html сторінки користувачеві. Він зображений на лістингу Б.5 додатку Б.

Метод home повертає головну сторінку сайту. На рисунку 3.28 зображено головну сторінку. Користувач системи може вибрати модель класифікатора зі списку. Також користувач може перейти до розділу документації, вибрати зображення для аналізу. При натисканні кнопки завантажити процес аналізу зображення ініціалізується. Така поведінка головної сторінки повністю відповідає діаграмі варіантів використання.

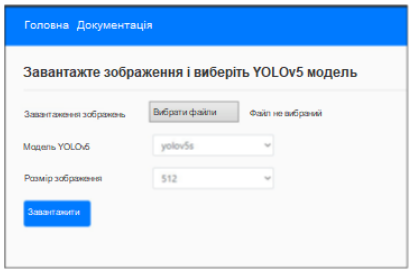


Рисунок 3.28 – Скріншот головної сторінки

На рисунку 3.29 представлені дві доступні для вибору моделі: YOLOV5s, YOLOV5x. Аналогічно користувач може вибрати розмір зображення. Процес відбитий на рисунку 3.30.

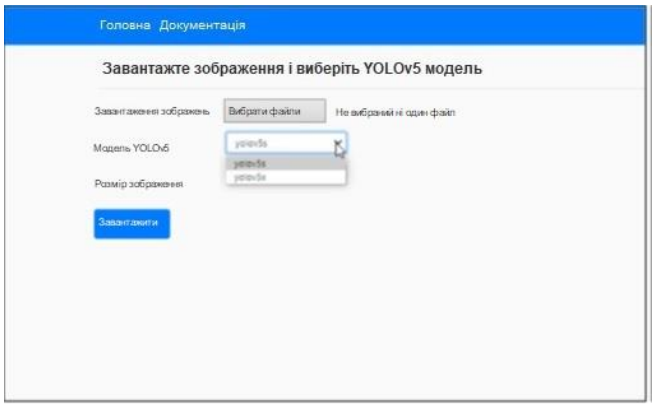


Рисунок 3.29 – Скріншот із відкритим списком моделей

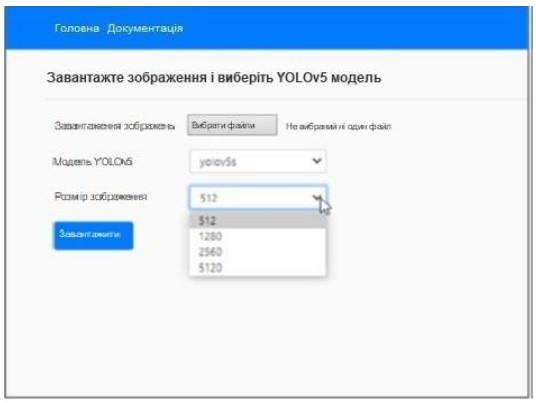


Рисунок 3.30 – Скриншот з відкритим списком множників зображення під час конвертації

Далі на лістингу Б.6 додатку Б зображено фрагмент шаблону home.html, відповідальний за секцію вибору розміру зображення.

Реалізовано два публічні Api методи, що відповідають вимогам системи. У додатку Б наведено лістинг Б.7 семантики методу для класифікатора.

На рисунку 3.31 можна побачити фрагмент сторінки документації, що описує поля AnalyzeItemsRequest.

Рисунок 3.31 – Фрагмент сторінки документації

Цей запит має валідації на кожне обов'язкове поле. Наприклад, у лістингу Б.8 додатку Б наведено перевірку імені моделі: вона повинна входити до переліку доступних для використання.

У запитах є обробник винятків, він зображений у лістингу Б.9 додатку Б.

Базовий клас, класи валідаційних та комунікаційних помилок були імпортовані з бібліотеки pydantic (див. лістинг Б.10 додатку Б). Тобто власна

реалізація не знадобилася.

Метод класифікації конвертує зображення у формат JPEG та створює масив із байт-кодів зображень, для подальшої обробки нейронною мережею. Ця логіка показана на лістингу Б.11 додатку Б.

При запуску сервера можна ввести ключ --precache-models, який викличе метод кешування моделей нейронних мереж. Цій логіці відповідає лістинг Б.12 додатку Б.

Аналогічно працює завантаження у методі класифікації, якщо не було вибрано кешування. У додатку Б лістинг Б.13 містить аналогічний код.

Після завантаження моделі вона викликається для обчислення результату.

У додатку Б лістинг Б.14 описує приклад методу класифікації.

Результат відповідає очікуваній відповіді системи (див. лістинг Б.15 додатку Б).

4 ТЕСТУВАННЯ ВЕБ-ДОДАТКУ

4.1 Функціональне тестування

Функціональне тестування відноситься до категорії тестування «чорної скриньки». Кожна функція програми тестується шляхом введення її вхідних даних та аналізу вихідних. При функціональному тестуванні програмний продукт, як правило, звіряється із зовнішньою специфікацією.

В рамках проведення функціонального тестування було здійснено ручне тестування основних функцій веб-додатку. Зміст тестів, і навіть результати їх виконання наведено у таблиці 4.1.

Таблиця 4.1 – Функціональне тестування веб-програми

Опис тесту	**Очікуваний результат**	**Результат проходження тесту**
Завантаження зображення. Вибрано та завантажується зображення з коректним розширенням .jpg або .png	Зроблено завантаження зображення. Ім'я файлу зображення відображено в інтерфейсі користувача	Тест пройдений
Завантаження зображення. Вибрано та завантажується зображення з некоректним розширенням	Завантаження зображення не зроблено. Відображено повідомлення про некоректне розширення файлу	Тест пройдений
Завантаження зображення. Зображення для завантаження не вибрано, але ініційовано процес завантаження файлу	Завантаження зображення не зроблено. Відображено повідомлення про необхідність вибору файлу для завантаження	Тест пройдений

Функціональне тестування ґрунтується на діаграмі варіантів використання. Усі тести, їх результат подані у таблиці. Таблиця 4.2 містить колонку «Примітка», в якій містяться покажчики до додатку із зображеним результатом після

проходження тесту.

Таблиця 4.2 – Результати проходження тестів, їх опису та примітки

Номер тесту	**Постановка тесту**	**Результат**	**Примітка**
1	Користувачеві доступна головна сторінка через один із підтримуваних браузерів.	Пройдений	Рисунок 3.28
2	Користувачеві доступна сторінка з документацією до сервера.	Пройдений	Рисунок В.2 додатку В
3	Користувач може вибрати фотографію для завантаження.	Пройдений	Рисунок В.3 додатку В
4	Користувач може вибрати модель нейронної мережі для класифікації елементів сторінки.	Пройдений	Рисунок 3.29
5	Користувач може вибрати розмір вихідного файлу.	Пройдений	Рисунок 3.30
6	Користувачеві повідомляється про відсутність обов'язкового поля.	Пройдений	Рисунок В.5 додатку В
7	Користувач може отримати результат про якість переданої сторінки за допомогою веб-сторінки.	Пройдений	Рисунок В.4 додатку В
8	Користувачу подано зображення з коректно розпізнаними елементами сторінки.	Пройдений	Рисунок В.4 додатку В
9	Користувач може завантажити файл результатів.	Пройдений	Рисунок В.4 додатку В
10	Якість оцінки відповідає у 80% випадків	Пройдений	Результат вірний у 83% випадків на вибірці 50 сторінок.
11	Сервер запущено	Пройдений	Рисунок В.1 додатку В

4.2 Інтеграційне тестування

Інтеграційне тестування – тестування, яке спрямоване на перевірку взаємодії між кількома частинами програми [25].

В рамках проведення інтеграційного тестування в систему через модуль інтерфейсу користувача здійснювалася завантаження зображень з перевірочного набору даних, які надалі оброблялися в модулі нейронної мережі. Далі отриманий результат оцінки інтерфейсу користувача відображався у вигляді інформаційного повідомлення користувачеві в модулі інтерфейсу користувача.

Інтеграційне тестування пройдено успішно, результати роботи модулів інтерфейсу користувача і нейронної мережі повністю відповідають міткам класів перевірочного набору даних.

4.3 Оцінка коректності роботи системи

На підставі результатів проведених тестів у рамках функціонального та інтеграційного тестування можна зробити висновок про коректність роботи системи.

Система повністю задовольняє функціональним та нефункціональним вимогам, визначеним на етапі проектування, а також забезпечує коректне представлення результатів своєї роботи.

Крім тестування виконуючого коду, також було проведено приймальне тестування (тестування якості) нейронних мереж.

Були проаналізовані випадкові приклади даних, наприклад, на малюнку зображено приклад нормалізованих вхідних даних для експертної нейронної мережі. Як бачимо, закономірність між кількістю класів і числами об'єктів класів дотримані (див. рисунок 4.1).

Також для класифікуючої нейронної мережі була порахована F-мера (див. рисунок 4.2) і закономірність між точністю і достовірністю (див. рисунок 4.3).

Оскільки прикладів, що містять спливаюче вікно не так багато, точність і частота знаходження такого об'єкта вкрай низька. Однак, якщо він є, то він є вагомим, тому що така поведінка була врахована при створенні тестових даних.

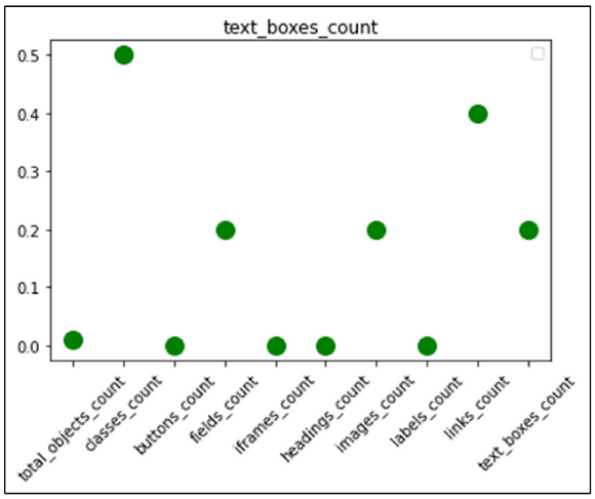


Рисунок 4.1 – Діаграма нормалізованих вхідних даних

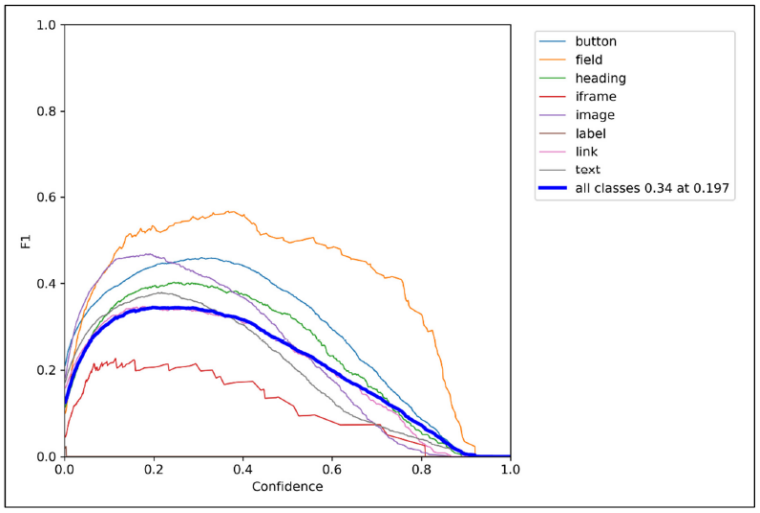


Рисунок 4.2 – Графік F-міри

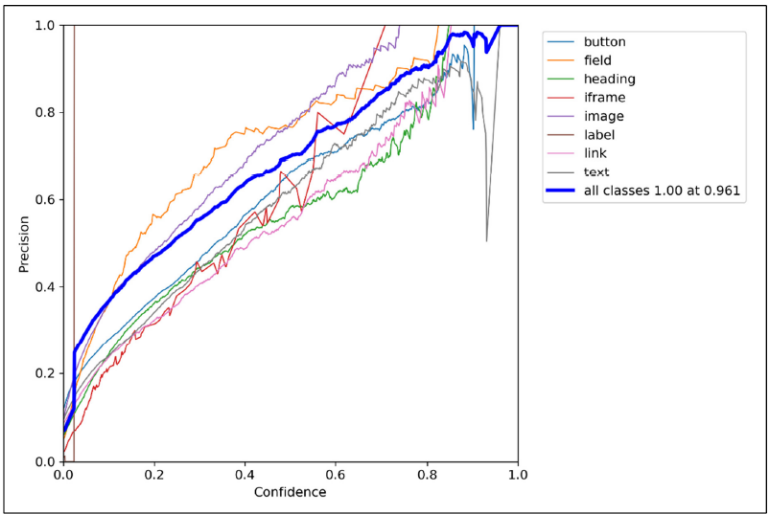


Рисунок 4.3 – Крива точності та достовірності

Для цієї мережі також важливо відобразити матрицю помилок. На рисунку 4.4 зображено матрицю помилок.

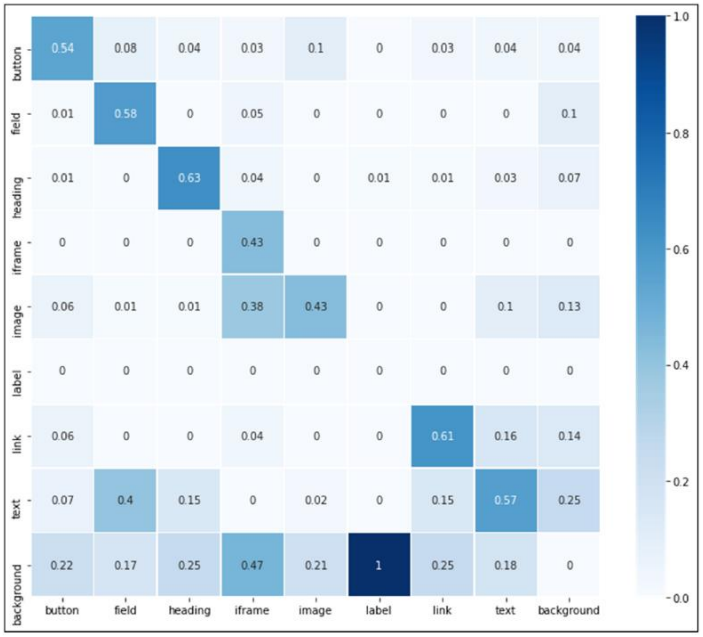


Рисунок 4.4 – Матриця помилок класифікатора

Причиною поганої класифікації класу label є те, що, як виявилося, розмічений датасет немає достатнього обсягу таких об'єктів.

Аналогічна крива точності та закономірності для експертної нейронної мережі була описана у розділі навчання нейронної мережі.

Іноді бувають аномальні результати. Наприклад, головна і не вирішена проблема: нейронна мережа хибно спрацьовує на сірий фон з дрібним малюнком і припускає, що це кнопка. Частота таких помилкових спрацьовувань не часто, але поширена. Найчастіше це впливає на вірність оцінки. Зміна мережі на YOLOv5x не дає бажаного приросту якості.

Експертна нейронна мережа була протестована з використанням 20 контрольних зображень різного призначення і очевидними недоліками і перевагами спроектованого інтерфейсу з точки зору різнорідності. Як було заявлено раніше, система працює з точністю 83%, це значення було підтверджено.

ВИСНОВКИ

Якість інтерфейсу користувача – це складна для оцінки сутність. Різнорідність веб-елементів один із важливих критеріїв, що становлять оцінку задоволеності. Цей критерій впливає простоту засвоєння інформації, сприйняття веб-сторінки, зручність управління системою. Актуальність теми підтверджується безліччю робіт.

В рамках виконаної роботи було розроблено веб-додаток для оцінки ієрархічності інтерфейсів веб-сторінок на основі аналізу скріншотів. Зокрема, було вирішено такі завдання.

Зроблено огляд існуючих методів оцінки юзабіліті інтерфейсу користувача та існуючих аналогів.

Здійснено аналіз вимог, розроблено метод оцінювання ієрархічності веб-інтерфейсів та здійснено проектування архітектури веб-додатку.

Досліджено предметну область і зробити порівняльний аналіз методів оцінки різнорідності інтерфейсу.

Розроблено методику оцінки різнорідності інтерфейсу веб-сторінок.

Спроєктовано архітектуру системи оцінки різнорідності веб-сторінок за допомогою нейромережевих технологій.

У подальших дослідженнь планується вирішення наступних завдань.

Розширення переліку ознак оцінювання ієрархічності.

Дослідження взаємозв'язку між експертною оцінкою та значенням метрики, що обчислюється.

Об'єднання модулів оцінки ієрархічності та різнорідності інтерфейсу веб-сторінок у єдину систему.

Крім практичного та дослідницького досвіду, в ході роботи було отримано безцінний досвід інтеграції навчених моделей нейронних мереж та клієнт-серверної програми. Були виконані завдання інтеграції нейронних мереж та їхнє практичне застосування.

Сама робота має подальший потенціал у розвитку: розширення методів

оцінки, вдосконалення поточного рішення, публікація сервісу та надання доступу до нього.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Stewart T. 2012. Websites – Quality and Usability // Behaviour & Information Technology, 2016. – Pp. 645–646.

2. How long do users stay on web-pages. URL: https://www.nngroup.com/articles/how-long-do-users-stay-on-web-pages (дата звернення: 15.05.2024 р.).

3. Asil Oztekin, Dursun Delen, Ali Turkyilmaz, Selim Zaim A machine learning-based usability evaluation method for eLearning systems. URL: https://www.researchgate.net/publication/256197830_A_machine_learning-based_usability_evaluation_method_for_ eLearning_systems (дата звернення: 15.05.2024 р.).

4. Amanatiadisa A., Mitsinisb N., Maditinos D. A neural network-based approach for user experience assessment. // Behaviour & Information Technology, 2014. – Pp. 321–333.

5. Márcio José Moutinho da Ponte, Antonio Morais da Silveira A Methodology for Evaluation the Usability of Software for Industrial Automation Using Artificial Neural Networks: Case Study—Eletrobrás. URL: https://www.researchgate.net/publication/224566946_A_Methodology_for_Evaluation_the_Usability_of_Software_for_Industrial_Automation_Using_Artificial_Neural_Networks_Case_Study_Eletrobras (дата звернення: 15.05.2024 р.).

6. DeLone W., McLean E. Information Systems Success: The Quest for the Dependent Variable. // Information Systems Research, 2012. – Pp. 60–95.

7. Noriaki K., Seraku N., Takahashi F., Tsuji S. Attractive Quality and Must-Be Quality // The Journal of the Japanese Society for Quality Control, 1984. – Pp. 39–48.

8. Courage C., Baxter K. Understanding Your Users: A Practical Guide to User Requirements Methods, Tools, and Techniques. – Morgan Kaufmann, The Morgan Kaufmann Series in Interactive Technologies, 2005. – 214 p.

9. Hassenzahl M., Tractinsky N. User Experience – A Research Agenda. // Behaviour & Information Technology, 2006. – Pp. 91–97.

10. Kincl T., Štrach P. Measuring Website Quality: Asymmetric Effect of User Satisfaction. // Behaviour & Information Technology, 2012. – Pp. 647–657.

11. Nielsen Norman. Group Articles & Videos. URL: https://www.nngroup.com/ articles (дата звернення: 15.05.2024 р.).

12. Palmer J.W. Web site usability, design, and performance metrics. //Information systems research, 2002. – Pp. 151–167.

13. QuickLinks label intranet. URL: https://www.nngroup.com/articles/ quicklinks-label-intranet (дата звернення: 15.05.2024 р.).

14. Navigation IA tests. URL: https://www.nngroup.com/articles/navigation-ia-tests (дата звернення: 15.05.2024 р.).

15. Popup problems. URL: https://www.nngroup.com/videos/popup-problems (дата звернення: 15.05.2024 р.).

16. Rules UX. URL: https://www.nngroup.com/videos/rules-ux (дата звернення: 15.05.2024 р.).

17. Effective online advertising. URL: https://www.nngroup.com/videos/ effective-online-advertising (дата звернення: 15.05.2024 р.).

18. Footers. URL: https://www.nngroup.com/articles/footers (дата звернення: 15.05.2024 р.).

19. Designing search suggestions. URL: https://www.nngroup.com/videos/ designing-search-suggestions (дата звернення: 15.05.2024 р.).

20. Better forms visual organization. URL: https://www.nngroup.com/videos/ better-forms-visual-organization (дата звернення: 15.05.2024 р.).

21. Grid layouts. URL: https://www.nngroup.com/videos/grid-layouts (дата звернення: 15.05.2024 р.).

22. Narayanan N., Balaji N., Jaganathan K. Deep Learning for UI Element Detection: DrawnUI 2020. – Sri Sivasubramaniya Nadar College Of Engineering, 2020. – 157 p.

23. Open Source of Roboflow. URL: https://roboflow.com/open-source (дата звернення: 15.05.2024 р.).

24. Datasets of Kaggle. URL: https://www.kaggle.com/datasets (дата звернення: 15.05.2024 р.).

25. Авраменко А.С., Авраменко В.С., Косенюк Г.В. Тестування

програмного забезпечення. Навчальний посібник. – Черкаси: ЧНУ імені Богдана Хмельницького, 2017. – 284 с.

26. Neural Network Models Effectiveness for Face Detection and Face Recognition / Smelyakov K., Chupryna A., Bohomolov O., Hunko N. // 2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), 22 April 2021, Vilnius, Lithuania. – P. 1-6. URL: https://ieeexplore.ieee.org/document/9431476

27. The Neural Network Technologies Effectiveness for Face Detection / Chupryna A., Bohomolov O., Ruban I. // IEEE Third International Conference on Data Stream Mining & Processing (DSMP), August 21-25. – 2020. – P. 201-205. URL: https://ieeexplore.ieee.org/document/9204049

28. Application of Neural Networks to Identify of Fake News / Onyshchenko Kostiantyn, Iryna Afanasieva, Vira Golian, Artem Khovrat // Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems. Volume II: Computational Linguistics Workshop. Kharkiv, Ukraine, April 20-21, 2023. Pp. 346-358. URL: https://ceur-ws.org/Vol-3396/paper28.pdf

29. Using Recurrent and Convulation Neural Networks to Indentify the Fake Audio Messages / Artem Khovrat, Volodymyr Kobziev // 2023 IEEE 7th International Conference on Methods and Systems of Navigation and Motion Control, MSNMC 2023 - Proceedings, 2023, pp. 174–177. URL: https://ieeexplore.ieee.org/abstract/document/10329236

30. Selection of Artificial Neural Networks for Disease Prediction / Iryna Kyrychenko, Oleksii Nazarov, Nural Huliiev, Oleksii Avdieiev // COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine (CEUR Workshop Proceedings (CEUR-WS.org)). Volume I: Machine Learning Workshop, pp. 236–248. URL: https://ceur-ws.org/Vol-3387/paper18.pdf

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ ЗА НАУКОВИМИ НАПРЯМАМИ КЕРІВНИКА ТА НАУКОВЦІВ КАФЕДРИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

26. Neural Network Models Effectiveness for Face Detection and Face Recognition / Smelyakov K., Chupryna A., Bohomolov O., Hunko N. // 2021 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream), 22 April 2021, Vilnius, Lithuania. – P. 1-6. URL: https://ieeexplore.ieee.org/document/9431476

27. The Neural Network Technologies Effectiveness for Face Detection / Chupryna A., Bohomolov O., Ruban I. // IEEE Third International Conference on Data Stream Mining & Processing (DSMP), August 21-25. – 2020. – P. 201-205. URL: https://ieeexplore.ieee.org/document/9204049

28. Application of Neural Networks to Identify of Fake News / Onyshchenko Kostiantyn, Iryna Afanasieva, Vira Golian, Artem Khovrat // Proceedings of the 7th International Conference on Computational Linguistics and Intelligent Systems. Volume II: Computational Linguistics Workshop. Kharkiv, Ukraine, April 20-21, 2023. Pp. 346-358. URL: https://ceur-ws.org/Vol-3396/paper28.pdf

29. Using Recurrent and Convulation Neural Networks to Indentify the Fake Audio Messages / Artem Khovrat, Volodymyr Kobziev // 2023 IEEE 7th International Conference on Methods and Systems of Navigation and Motion Control, MSNMC 2023 - Proceedings, 2023, pp. 174–177. URL: https://ieeexplore.ieee.org/abstract/document/10329236

30. Selection of Artificial Neural Networks for Disease Prediction / Iryna Kyrychenko, Oleksii Nazarov, Nural Huliiev, Oleksii Avdieiev // COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine (CEUR Workshop Proceedings (CEUR-WS.org)). Volume I: Machine Learning Workshop, pp. 236–248. URL: https://ceur-ws.org/Vol-3387/paper18.pdf